Systems Security Engineering

Final Technical Report SERC-2010-TR-005

Principal Investigator: Jennifer Bayuk, Stevens Institute of Technology

Team Members

Dennis Barnabe, NSA/ESEA

Jonathan Goodnight, OUSD(AT&L)/DDRE/SE

Drew Hamilton, Auburn University

Barry Horowitz, University of Virginia

Clifford Neuman, University of Southern California

Stas' Tarchalski, Stevens Institute of Technology

Contract Number: H98230-08-D-0171 , DO 001, TO 0002, RT 008

Report No. SERC-2010-TR-005

August 22, 2010

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED
22 AUG 2010		**00-00-2010 to 00-00-2010**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Systems Security Engineering	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stevens Institute of Technology,Castle Point on Hudson,Hoboken,NJ,07030-5991**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**78**	

This page intentionally left blank

System Security Engineering

A Research Roadmap

Table of Contents

1. Executive Summary 7
2. Problem Statement 8
3. Solution Criteria 11
4. Proposal 20
4.1. Security Definition 21
4.2. Frameworks 22
4.3. Metrics 23
4.4. Workforce 25
4.5. Systems Engineering Methods, Processes and Tools 26
4.6. Advanced Research Topics 27
4.7. Coordination 28
5. Summary and Next Steps 28
6. Contributors 29
Appendix A: Additional Detail on Selected Research Modules 33
Security definition (Reference Section: 4.1) 34
A. Security Standards Reconciliation 34
B. The Utility of Security Best Practices 35
C. Security Policy Compliance 36
D. Adaptation of Security Policy and Mechanism 37
Security Frameworks (Reference Section: 4.2) 38
E. Critical Program Information Protection 38
F. System of Systems 39
G. Configuration Hopping 41

H. Continuity of Communications 42

I. Data Continuity Checking 44

J. Denial and Deception 46

K. Shared Command Information Sharing 47

L. Physical Security Frameworks 48

Security metrics (Reference Section: 4.3) 49

M. Architecture Metrics 49

N. Risk Metrics 51

O. Security versus Convenience 52

P. Security Trade Spaces in Emerging Technologies 54

Q. Trust Assessment Models 55

Security workforce (Reference Section: 4.4) 58

R. Workforce Education 58

S. Security Requirements Process 59

T. SE Career Path 61

Security MPTs (Reference Section: 4.5) 61

W. Exploring Nearby Disciplines 61

X. BKCASE Security Section 62

Security advanced topics (Reference Section: 4.6) 64

Y. Agile Architecture 64

Z. Executable Architecture 65

AA. Critical Functionality 67

Security Research Coordination (Reference Section: 4.7) 68

BB. Coordination 68

CC. Hypothesis Test 68

Appendix B: Glossary 70

Appendix C: SERC Security Research Workshop Agenda 73

Appendix D: References and Bibliography 75

This page intentionally left blank

System Security Engineering

A Research Roadmap

1. Executive Summary

The US needs dramatic improvements in systems security. Current defensive strategies, based principally on strengthening system peripheries, inspections, and similar bolt-on techniques add tremendously to cost and do not respond effectively to the growing sophistication of attacks. Systems cannot be assumed to have static boundaries, static user communities, or even a static set of services. To a great extent, systems engineers are inadequately prepared to address system security requirements.

The failure of traditional systems engineering methods to address system security issues is due to the fact that these methods rely heavily on requirements gathering and modeling. In the realm of security, requirements gathering has been influenced by the fact that a variety of industries have developed system security standards. These have been presented to systems engineers as complete system security requirements, when in fact they cover only basic technology control measures. In the realm of security, engineering models are based on assumptions that a system is bounded by technology and that off-the-shelf technology control measures can be configured in combination to adequately address most security requirements. This is a false assumption.

However, simply removing these assumptions and challenging the systems engineer to put aside security standards and models and start afresh will not resolve systemic security problems. The existing standards and models came about because security is a difficult problem to address. Current standards and models have been embraced by a generation of practitioners who entered the systems security field over the past forty years because those practitioners found common solutions to diverse security problems and shared them. This work is significant and should be leveraged by integrating it with a fresh look at the mission of the systems engineer with respect to security.

This document establishes a research roadmap for System Security Engineering.

> *Systems Security Engineering (SSE) is defined as an element of system engineering that applies scientific and engineering principles to identify security vulnerabilities and minimize or contain risks associated with these vulnerabilities [1].*

This roadmap methodically extracts the value of current approaches to systems security and integrates it with a systems thinking perspective. This path is expected to result in systems designs that shield against commonly known attacks, provide cognizance of changes in the threat environment, and are resilient in the face of unexpected attacks. This outcome requires that we are able to efficiently apply security standards, that we improve intelligence gathering capabilities that are relevant to a system's mission, and that we consider mission assurance a core system capability going forward. In order to do this, a systems engineer will need a clear definition of security, and a way to compare security metrics to other capabilities in the system trade space.

Although this roadmap does draw on existing security standards and processes, it makes no assumption about the utility of historical methodology. Rather, it brings a scientific approach to the study of systems security engineering. By applying empirical scientific methods to the problem of security, it will establish firm evidence that research results will effectively address systemic issues going forward. A rigorous academic approach to a problem has the following characteristics:

- clear **problem statement**
- thorough problem **background** description including a full literature review
- clearly defined **solution criteria**
- **proposed hypothesis** formulated to shed light on a solution and how it may be (dis)proven
- **summary** of contributions to field **and** a statement of **next steps**

The bolded words in the bullet list above label the remaining sections of this report. The sections that follow describe how the SERC Security Engineering team used this approach to build a systems security engineering roadmap as well as how this academic approach informs the research recommended in the report. The roadmap is composed of a set of research modules with both short term and long term goals. The short term research modules will lay the groundwork that becomes a launch point by which longer-term milestones may be accomplished. The common element in both short-term and long-term goals is to produce useful and viable methods, processes, and tools (MPTs) for systems engineers to identify and reproduce architectural patterns for systemic security.

2. Problem Statement

Security MPTs have emerged over time in response to new threats and risks to enterprise assets. Physical security MPTs have evolved over time to protect facilities and installations as well as to detect or deter physical harm. Computer and communications security MPTs has similarly evolved. Protection and detection measures have been systematically applied to electronic information, both network and locally accessed. The ever-increasing level of the cyber dimension to physical systems, including physical security systems, has made cybersecurity the main focus of systems security research. As the number and complexity of different types of security threats and risks have grown, systems security MPTs have

grown correspondingly complex. Since 1997, the lack of a coordinated systems strategy has been repeatedly identified as a subject requiring a national research agenda [2]. The first such agenda was formally documented in 1999 [3, 4]. The most recent was published in 2009 [5]. These research strategies tend to concentrate on hard problems in systems security. The challenge is meant to mirror the gauntlet of canonical hard math problems presented by Hilbert to the International Congress of Mathematicians in Paris in 1900. None of the problems are expected to be immediately solved, but rather to be studied by everyone as the path to advancing the profession. The current hard problems in systems security are:

- Scalable trustworthy systems
- Enterprise-level metrics
- System evaluation life cycle
- Combating insider threats
- Combating malware and botnets
- Global-scale identity management
- Survivability of time-critical systems
- Situational understanding and attack attribution
- Provenance of information, systems, and hardware
- Privacy-aware security
- Usable security

This document does not attempt to recreate or redefine the list of hard problems in systems security research. Today's systems engineers often do not even use today's security MPTs effectively.

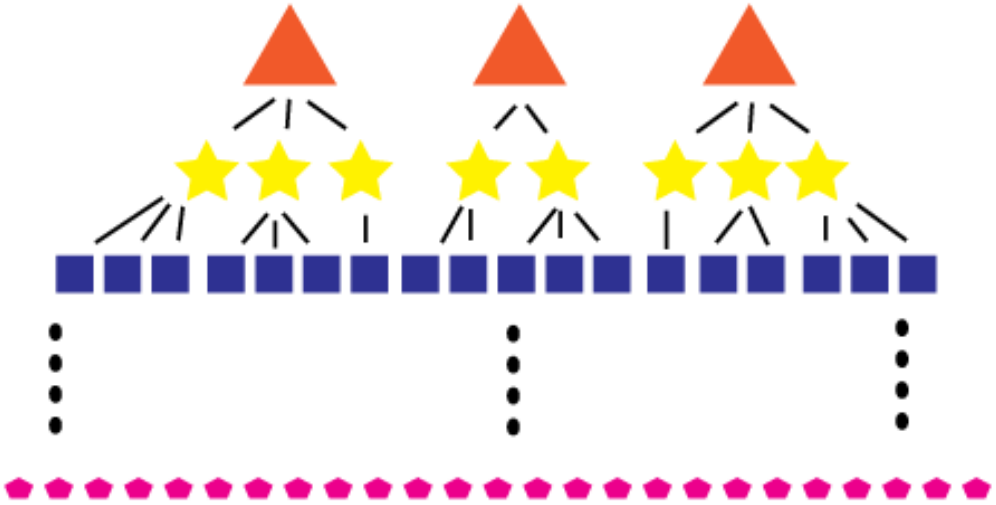
> **The problem statement for this research agenda is to evolve the practice of SSE to make effective use of the available security technology, that is, to advance the state of the art in SSE MPTs.**

The security research community's emphasis on hard problems is presented to establish the unfortunate truth that even if governments and businesses decided tomorrow to spend the money it would take to secure cyberspace, there is no clarity on what should be on the shopping list. Although security MPT standards and best practices have been accumulating for decades, malicious activity in cyberspace is not thwarted simply by application of those standards. Rather, cyber-perpetrators utilize the same cyberspace services that are available to those who are authorized to use them. The goal of a cyber-intruder is rarely to damage a system, but to exploit it to gain objects of value. Cyber-incidents of espionage and fraud are more common than cyber-terror. Cyber intruders study our security standards in order to avoid the defenses bases on them as they move seamlessly through our systems masquerading as authorized users.

Systems vulnerabilities that are prevalent in cyberspace are exacerbated when software is embedded in hardware components. Though standards for security components have been established, from the

point of view of the SSE, compliance with these standards may as well be included in the hard problems list [6, 7]. A 2009 report on Trusted Defense Systems identified significant vulnerabilities in mission-critical functionality due to the ability of adversaries to corrupt technologies, introduce malicious code into the supply chain, and otherwise gain access to the military systems and networks [8]. As depicted in Figure 1, any program of record will have layers of components that are subject to these threats, and each level introduced a point of abstraction through which these threats may be obscured. SSE must focus on components that are highly critical to the success of the programs and adopt MPTs that allow verification and validation at appropriate interfaces and lower-level components that allow the attribution of security at the higher, mission assurance level.

Figure 1: Functional Decomposition



When systems engineers are provided with requirements that have to do with core systems capability, the response is a discussion of trade-space alternatives. For example, if more performance is required from a ship, an engineer may suggest an additional engine, a replacement engine, a reduction in load, a reshaping of the keel, or a variety of other alternatives that have an associated trade-off with existing ship capability. In order to get an additional engine, there must be a reduction in space in the engine room, and perhaps reinforcement of the floor. In order to get a reduction in load, some other heavy material must be removed from the boat. In addition, each alternative will be associated with some cost.

In contrast, when systems engineers are provided with a security requirement, the requirement is often not clearly defined in terms of capability. Instead, it is often defined in terms of technology. Security is required in order to be in compliance with some set of standards that are translated into technical requirements. Especially in the realm of cybersecurity, the requirements are not stated in terms of capability but rather as configuration. For example, “the system authentication portal shall be protected by a firewall,” or “there shall be a central repository for user identity administration.” This type of

requirements puts a systems engineer in a position of order-taking as opposed to negotiation. There is an assumption that the existing system capabilities will not change. Without any flexibility in adjusting other system capabilities to achieve the security capability, an engineer will revert to the existing set of readily available solutions to the technology configuration requirement, decide how to most easily integrate them, and present the cost associated with each implementation alternative.

As is evident from cases in which security standards were followed yet systems were breached, system compliance with security standards is not an adequate metric by which to judge whether a system is secure [9]. Securing cyberspace, or even one system's corner of it, is a complex problem without a current solution. Nevertheless, this roadmap attempts to establish confidence in the ability of systems engineering methodology to support a structured approach to the determination of whether a system may be considered to be secure in the context of its mission or purpose. Moreover, current methods of collecting security configuration metrics may be useful in the process. The knowledge gap lies in the ability of current SSE MPTs to properly assess whether chosen security controls were appropriately selected, given system security requirements. This research will fill that gap by strengthening current capability to assess security with respect to system requirements. Groundwork had been laid for using systems thinking as an approach to security architecture issues. Recent work by various research communities both within and outside the SERC proposes using systems thinking concepts as a method for improving the quality of security engineering efforts [10]. The research proposed in this roadmap will extend these efforts into architectural approaches for consideration in the design phases of systems engineering process. It will inform functional decomposition and fundamental design to ensure that security solutions are embedded into system design. It will also provide the systems engineer with MPTs for valuing security solutions that can be used by stakeholders to make risk-based tradeoffs between security and other system capabilities. It will provide models and techniques for verification and validation of security requirements.

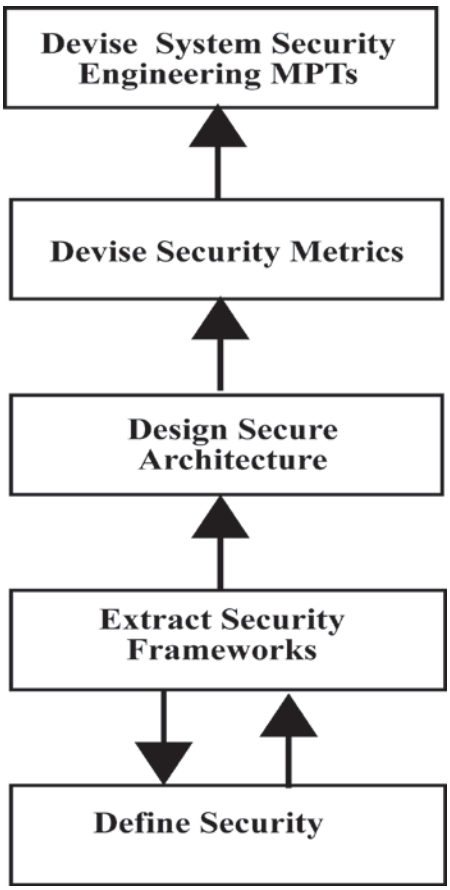
3. Solution Criteria

Any solution will directly reflect our problem statement, and thus, our goal is to improve the security effectiveness of systems engineers. Criteria of such a solution will be the ability of systems engineers to function effectively when faced with security requirements. There are several factors that present obstacles to this goal. For example, as previously discussed, compliance with security standards and best practices are frequently mistaken for security capabilities. In order to break away from the thought patterns of security that are well established in the mind of the professional systems engineer, a new paradigm for systems thinking with respect to security must be established. Such a paradigm would have to depict security as a completely tangible concept. A decomposition of our problem statement has led to a list of systems engineering security capabilities that are individually well-defined, and may be combined to demonstrate value in addressing the overall problem of evolving system security engineering practice. These capabilities are:

- Security Definition – A way to describe systems security that helps to clarify the scope of efforts that system engineers should pursue in their efforts to identify and address security problems
- Security Frameworks – A framework is an abstraction of the systems context with respect to security that can provide the basis for classification of both systems security architecture and associated security solutions. It provides a way to map enterprise asset landscapes to threat landscapes in order to quickly identity system security requirements and test potential solutions.
- Security Metrics - Measures of security effectiveness.
- Systems Engineering Workforce Development – Innovative ways to improve the current proficiency of the security engineering workforce and to align the security definition, framework, and metrics research area with other systems engineering fields to facilitate awareness of its importance to overall system resiliency, and enable contributions by a larger class of stakeholders.
- Systems Engineering Methods, Processes, and Tools – MPTs are needed to guide the security specification activities of non-security specialists in the workforce. These tools must take the coarse-grained specifications of specialists and create constraints that will be directly enforced by the security technologies implemented by security specialists, and by other components of the system such as the operating system and network components. These tools must be usable by non-specialists and must capture the expected behavior of the control flows, information flows and availability characteristics of the system. The ability to enforce such constraints collected by these tools will affect the architecture and framework aspects of this roadmap.
- Advanced Research Topics – Research intended to produce out-of-the-box systems thinking and leap-ahead security architectures.

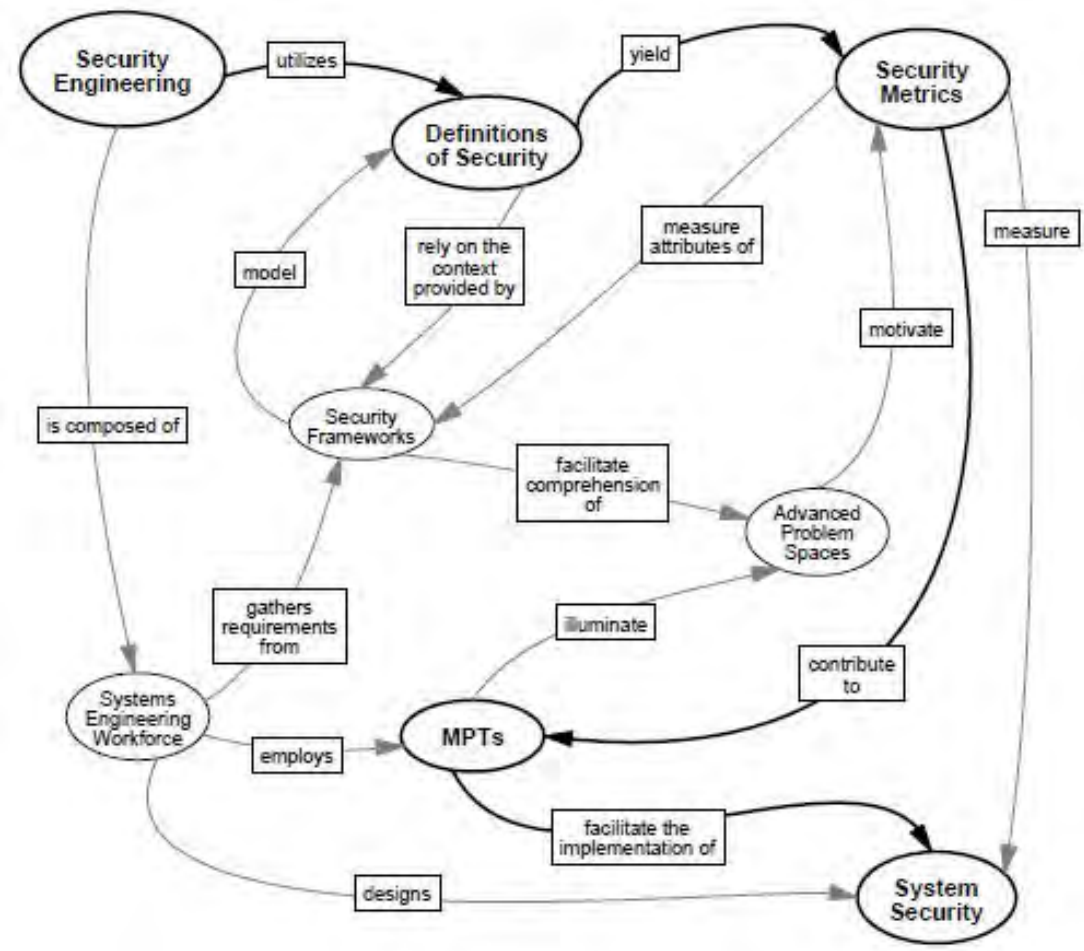
Figure 2 demonstrates that the capabilities to be addressed by any problem solution are interrelated. Most importantly, there must be a definition of security that allows comparison with systems attributes. Without a clear definition, SSE efforts have no clear goal. However, systems attributes that enable or allow for emergent security will be different depending on the mission of the system and the context in which it operates. Hence, it is important to identify that mission and context as a framework within which to understand the definition of security. That clear understanding should allow the design of alternative security architectures, as well as metrics that can be applied to those architectures in order to determine their effectiveness in maintaining system security. These metrics will then play a key role in the development of new MPTs for systems engineering.

Figure 2: Capabilities Required to Improve SSE



The capabilities required to improve SSE are related as illustrated in Figure 3. The figure shows the relationships between the capabilities, and thus provides a conceptual foundation for the research roadmap. The capabilities are depicted using a systems engineering job aid, a *systemigram* [11]. A systemigram is read from left to right, top to bottom. Circles contain nouns, which may be objects or concepts. Lines are called threads, which link the nouns. A systemigram describes a system identified in the top left corner succinctly by way of a "mainstay" thread, which connects the system to be defined with its main function or purpose, identified in the bottom right corner. The mainstay is a high level process description that is generally agreed by those who best understand the system. Other threads describe actions taken by the system that, though not central to its purpose, are nevertheless associated with any system so named. A systemigram does not produce a single paragraph of text, many of its threads skirt around its subject in an effort to add dimensions to the definition. The mainstay thread may be viewed as the core definition. But there is no assumption that the mainstay can stand on its own.

Figure 3: Security Engineering Research Roadmap Systemigram



The mainstay thread in Figure 3 is a bold line. It depicts the systems engineering security roadmap itself as a system whose primary function is to produce system security. The conceptual utility of the systemigram is also evident in that the secondary relationships between nodes on the mainstay, as well as the secondary threads, which are equally important to understanding the research roadmap as a whole. From the threads on the left, it is clear that stakeholders relies heavily on the systems engineering workforce to design security into systems, and that this workforce leans heavily on security frameworks for requirements, and on methods and tools to reduce the complexity of advanced problem spaces into comprehensible and measureable security features. The figure shows that security engineering utilizes definitions of security that yield security metrics that contribute to MPTs that facilitate the implementation of systems security. Those definitions rely on the context provided by security frameworks that facilitate the comprehension of advanced research topics.

This roadmap description reflects the reason that our problem statement is important. The research roadmap should not be concerned with the security effectiveness of systems engineers if the ultimate result is not more secure systems. The roadmap should create SSE MPTs that transform current systems engineering practice.

Other threads in the systemigram illustrate complementary aspects of the SSE roadmap. It is intended to focus security research on solutions to problems faced by systems engineers. For example, a security research module in the area of security metrics approached in the context of this roadmap would be required to clearly state the definition of security with respect to the framework of the system on interest, as well as to demonstrate the utility of the metrics to contribute to MPTs that will facilitate the implementation of secure systems. Currently, there are several examples of security metrics research literature in which the utility of the candidate metrics to support MPTs is not obvious. These include security metrics for mathematical modeling of security management processes [12], weighting network forensics evidence to increase probabilities of conviction [13], quantifying threat surface using hidden Markov models [14], using game theory to determine security investment strategies [15], and complex mathematical models for assessing software security [16]. Most of these are the subject of one or two papers by the same group of authors, and rely on data that is not completely described (and also usually include subjective measures of probability). By focusing security research on the common goal of assisting the systems security engineer, it is envisioned that multiple independent research efforts may be more comprehensible and potentially composable in the creation of MPTs.

Roadmap research results are expected to reorient the systems engineering workforce by assisting in the interpretation of security standards and the production of security requirements, while motivating the development and application of security metrics. These metrics contribute to the development of methods and tools that can be used not only to measure or assess, but also to facilitate the implementation of system security. The systemigram reflects the systems thinking perspective of the roadmap research team. Others may argue about which path through nodes via links should be taken as the mainstay, or disagree as to the central purpose of SSE, but if there is any question about whether the overall systemigram accurately depicts the problem space of the systems security engineer, it may be concluded that they are describing a different problem than the one addressed in this roadmap.

In preparation for defining this roadmap, MPTs related to the first four capabilities in the bullet list above Figure 2 (Security Definition, Metrics, Frameworks, and Workforce) have been methodically explored via literature surveys [17]. The result yielded only systems security standards and associated practice aids. Perhaps the most germane discovery from this effort was a draft ISO standard on *Systems and software engineering, Systems and software assurance* [18]. Created by systems engineers for systems engineers, this guidebook adopts the perspective that system security is the justified confidence that the system functions as intended and is free of exploitable vulnerabilities. Though it does not define security, it provides a common vocabulary on *assurance* from a systems engineering perspective, and outlines a process by which it may be pursued. Because it is vitally important to the success of this roadmap that research results be understood and adopted by systems engineers, this guidebook should

be used as a launch point for further guidance-oriented publications. Any solution is expected to fully utilize all current guidance that makes sense while at the same time changing the state of the art in SSE from a process-oriented to an outcome-oriented approach.

If a complete set of features that fully met the definition of security fully was transparent to a systems engineer, then there would also presumably be a way to measure the extent to which it was implemented. The goal is an ordinal-or-better security metric. However, it is generally not possible to define security outside of the context of system operations and threat landscape, so the framework within which a system operates will have relevance to both the definition of security and the way it is measured. This roadmap therefore places heavy concentration on modules devoted to security definition, metrics, and frameworks as a prerequisite to the production of new MPTs for SSE.

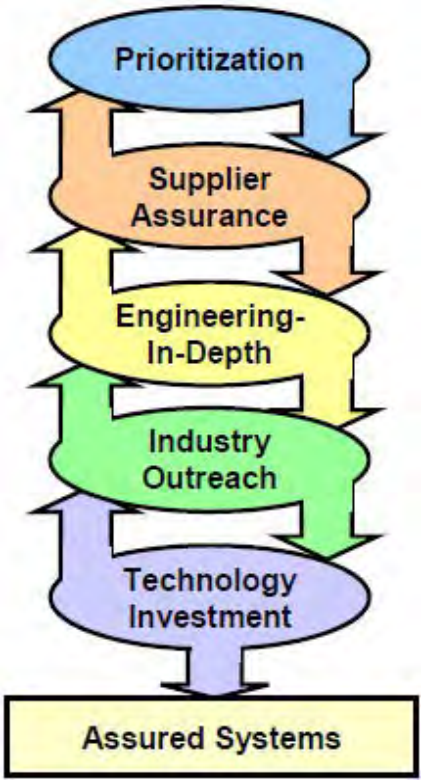
With comprehensive definitions of security, security metrics, and cohesive conceptual frameworks, advanced research topics in security architecture may be evaluated for efficacy in solving systemic security problems. Moreover, research into these architectures is a long-term prospect that should not wait for all definitional foundations to be built. Many security engineering researchers have so much experience in the multiple ways in which security can be defined and measured, as well as the utility of standards and security problems faced by systems engineers, that they should be launched on research programs that may provide the leap-ahead and paradigm shift that are thought by many to be essential to the solution of today's complex security problems [19]. Also note that the same set of researchers working within a specified framework may simultaneously address several of these fundamental systems security research requirements.

Our solution criteria should also specifically address the needs of warfighters and intelligence gathers for secure systems. Using the concepts and MPTs that would result from this program, systems engineers will be better able to think through and resolve hard trade-offs where systems security features are in the trade space. This research program does not specify the properties of systems but how to engineer capability to operate in an environment of ever-changing threats. The research should range from program information criticality analysis to quantifying the cost of protection countermeasures.

Figure 4 is a vision of success for a systems engineering directorate within which this research program is expected to operate. It illustrates that the ultimate target for systems engineering security efforts is systems assurance. In the context of the directorate, requirements for system security are defined in terms of assurance, and must be fulfilled by distributing resources in the security trade space properly among systems and their critical components. Diligence should be exercised to ensure that components are not vulnerable to supply chain risks. System designs must incorporate capability persistence at a known level of assurance. Known assurance levels are also facilitated through cooperation with key commercial component providers. This security-aware systems engineering approach results in technology investment that significantly and positively impacts systems' ability to detect and mitigate system vulnerabilities.

In order to provide security engineers with the MPTs they need to operate effectively in the context of Figure 4, we must first equip them with a way to define *system assurance*. As security has connotations ranging from sociological and political to technical, systems engineers faced with security requirements need a firm foundational knowledge on which to base an opinion as to which systems features fulfill systems security requirements. Providing this knowledge is therefore included in our solution criteria.

Figure 4: Vision of Success[1]



One finding from the roadmap team's literature surveys is that the definitions of *security* are fairly consistent across existing systems security standards. Figure 5 displays those definitions in the form of a systemigram. The mainstay of the systemigram in Figure 5 is a definition systems security, which has been defined the systemigram of Figure 3 as the purpose of SSE. It is a stake in the ground on what is meant by systems security.

[1] Vision of Success figure was presented by Kristen Baldwin at the SERC Security Engineering Workshop on March 31st 2010.

Figure 5: Mainstay of Security Systemigram

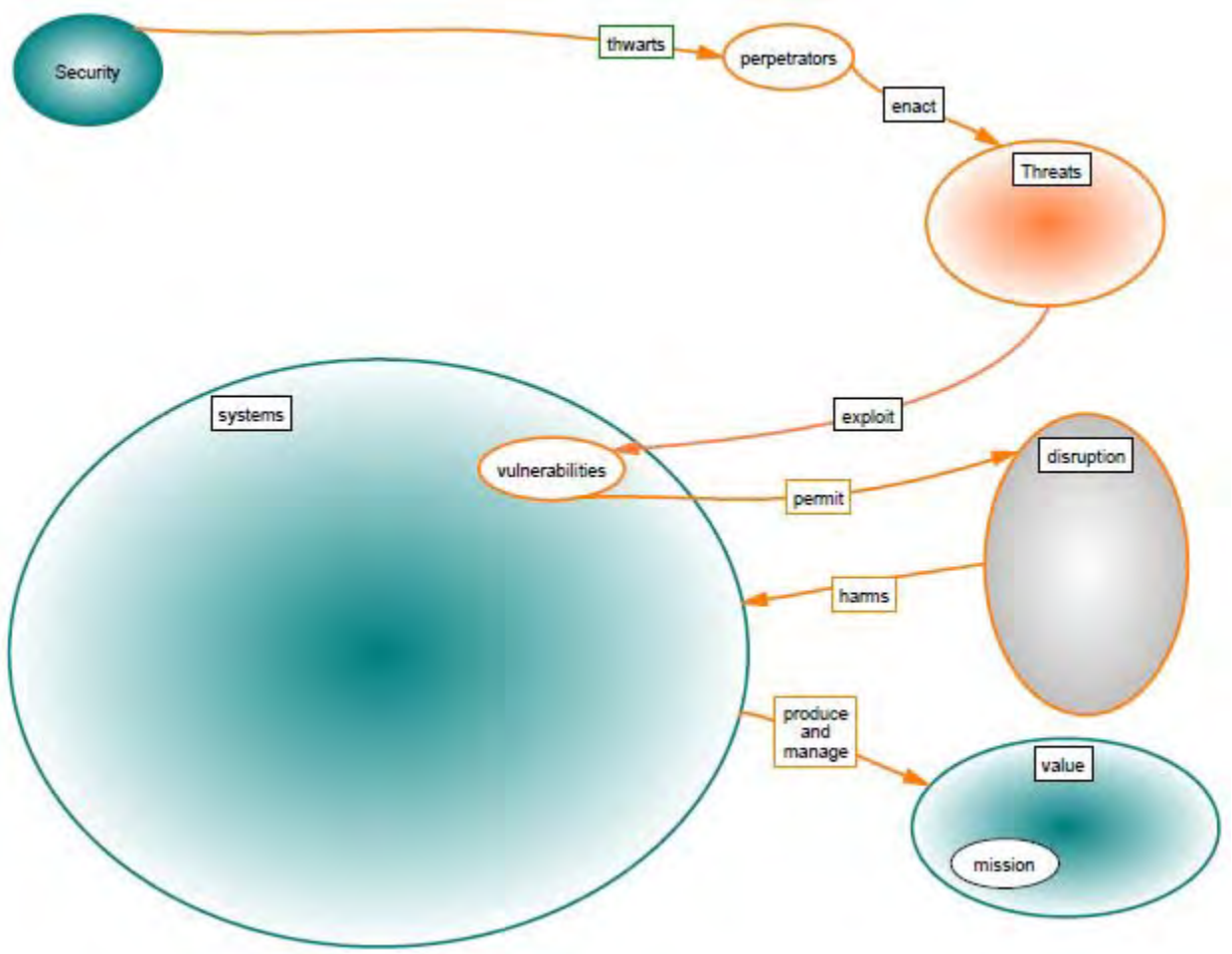
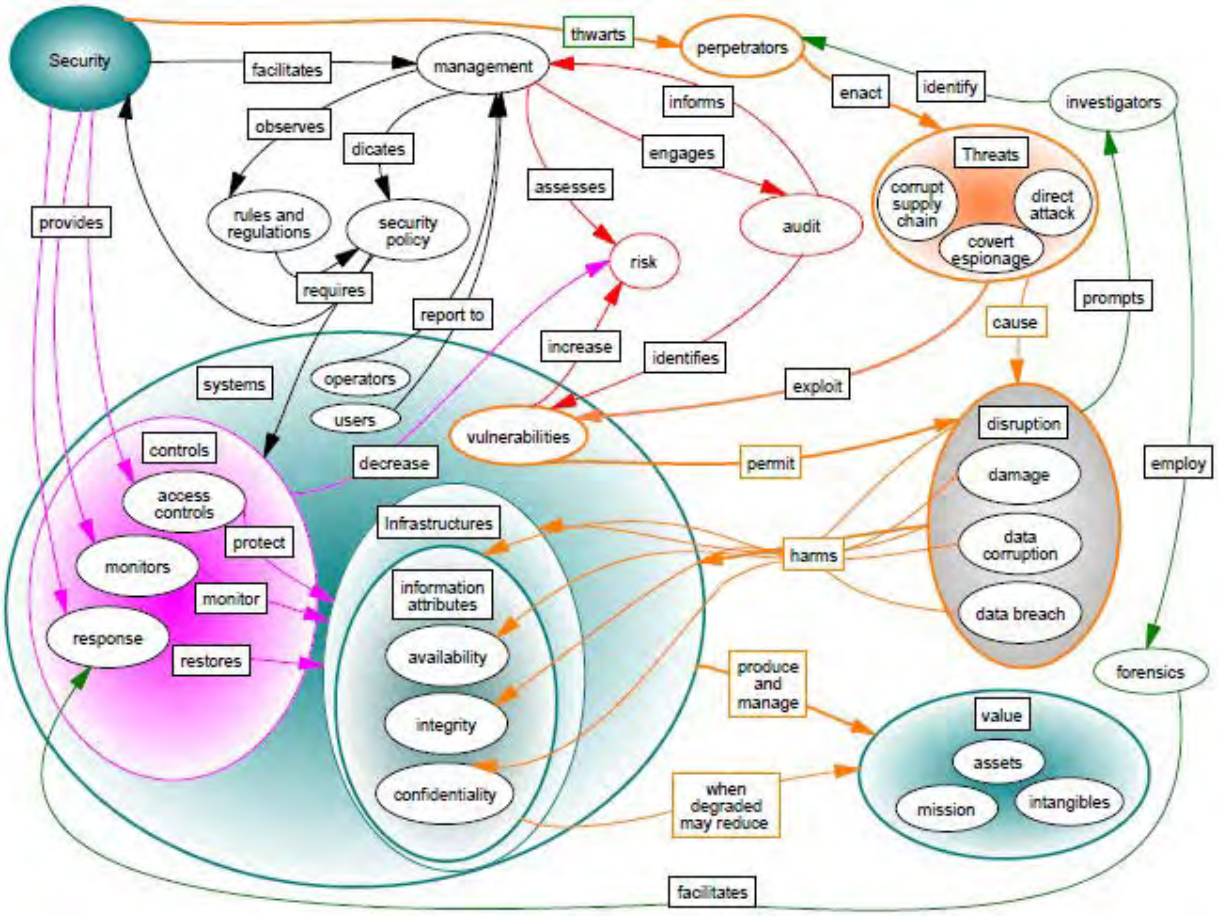


Figure 5 defines security as something that thwarts perpetrators who enact threats that exploit vulnerabilities that damage systems that produce and manage value. Mission is mentioned by some standards. Others mention value in terms of assets or intangible system attributes such as reputation. Figure6 shows the depiction of security that includes aspects of its definition other than that of mainstay thread. Though complex, it contains enough commonality that researchers should be able to identify where their contribution fits in. One of the security research modules is expected to be to refine this definition of security and validate that it covers all industry standards. It would then become a model-based tool to help systems engineers understand security standards. Figure 6 is thus considered a draft in progress.

Figure 6: Full Draft Security Systemigram



Hence, a key element of the roadmap is to provide capability for systems security researchers to self-assess the value of a potential contribution to the field of SSE, and/or to clarify some aspect of systems security itself. There is no expectation that simply because an examination of security standards currently produces a definition of security that looks like Figure 6 that this definition is entirely appropriate to model the systemic security properties of today's systems. Minimally, this figure would depict how today's security standards illustrate security and provide a launch point for improvements in the definition that will make the utility of research results more transparent than they have been in the past.

It is expected that there will be coordination and oversight efforts to ensure that research in individual modules is coordinated and inform each other. This work will require subject matter experts to review research results and validate their applicability to other research modules. It will also require coordination with research organizations and systems engineering associations outside of the SERC to ensure that the SERC research modules make the best available use of current research results. This

oversight function should also periodically survey systems engineering practice in an effort to determine whether security effectiveness has been improved. Hence, we envision a research module that will track correlations among the complete set of research, and even lead a wide variety of investigators on various research areas to meet and share ideas, and/or to attend workshops or conferences run by like-minded research organizations.

4. Proposal

Following the rigorous scientific approach to problem solving described in the executive summary, the proposal is in the form of a hypothesis that is: If a program were launched to enhance the capability of the systems engineer to design secure systems, then systems security in general will be improved. The reference to the program in hypothesis is to the research outlined in this section. Embedded within this hypothesis are assumptions that each research module recommended by the program will achieve some subset of the capabilities specified to be addressed in the discussion of Section 3 on solution criteria. Following the academic rigor of the overall roadmap, each of these modules is defined as an empirical study. More detail on the systems thinking behind each research proposal has been included in Appendix A. The final research module, 4.7.2, is a formal test of this hypothesis.

Note that there is any number of additional research modules that would meet criteria for inclusion in this roadmap, and that those included in Appendix A are meant to be representative of the type of research that would contribute to the goal of providing the capability described in the corresponding subsection. For each research module in the appendix, we include a problem statement, some background, solution criteria, and a proposed hypothesis. The descriptions are uniform to the extent required to preserve overall document readability.

Within the overall goal of enriching the security capability of the systems engineer via enhanced MPTs, some of the modules are related by a common purpose. They also may be further related in that the results of some modules are expected to inform some others. The overall structure of the roadmap relies on at least partial results of some research modules to inform others. For example, security frameworks emerge from examining use cases for security in different environments. There are also many existing security frameworks in the form of published standards documents. Both types of frameworks will be useful in establishing a working definition of security. These working definitions will further illuminate the security requirements presented via frameworks, and secure architecture will be designed to meet requirements. Where architecture and requirements are known, security metrics may be specified.

Just as outputs from different research modules are expected to inform each other, individual research modules would be expected to take advantage of endeavors with similar objectives within communities of researchers and practitioners engaged in similar work in various other endeavors. To ensure that the roadmap meets its solution criteria, it expected that research results and collaboration opportunities included in the *coordination* module would be accomplished in close consultation with program sponsors.

The remainder of this section contains one subsection for each capability required to improve SSE that was identified in Section 3 as solution criteria. These subsections describe the overall approach to research within the security engineering capability. Appendix A includes a list of research modules proposed for each capability. Each module description is described in a table that summarizes the problem statement, background, solution, and next steps. The table also includes the expected timeframe and dependencies of the research effort. The timeframe for each individual module is the expected timeframe in which research results may be practically be achieved. The dependencies indicate that they are expected to be informed by modules preceding it. Note that a module with dependencies may be selected for a research task even though a module on which it is dependent is not. However, such a selection may add to the expected timeframe in which research results are expected to be achieved. Actual research tasks inspired by this roadmap may of course contain other elements of interest to sponsors that would affect scope as well as expected timeframe.

4.1. Security Definition

This research is focused on clarifying the goals and objectives of security engineering endeavors. It is expected to include a comprehensive examination of established security standards and best practices relevant to systems engineering. It will compare these best practices to each other and to the systems for which they are primarily targeted. It will examine the utility of the standards in supporting the decision-making capability of systems engineers. The result of this research thread will be outcome-based approaches to SSE that start with clear definitions of the goals and objectives of SSE efforts.

As depicted in Figure 2, results of Definitions modules would both provide firm foundation and inform all other research modules. Documentation resulting from these studies will utilize concepts and vocabulary from the *ISO/IEC DTR 15026-1, Systems and software engineering — Systems and software assurance* [18]. This vocabulary described systems assurance as “the justified confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle” [20]. MPTs produced by this roadmaps will this form a consolidated framework and guide for engineering interpreting these standards. The definition modules will produce an intuitively comprehensible model of thinking about security standards that will allow systems engineering to make use of the best features of all available standards without requiring them to read multiple checklists of considerations that may not apply to their system of interest. The output of these modules would be an alternative media for representing systems security thinking, which itself would be expected to evolve into a systems engineering security standard.

The literature survey performed for this capability revealed that the definition of security is fairly consistent across existing systems security standards. We also found that these definitions are capable of being modeled in the form of a systemigram. A common model of security that incorporates common definitional attributes and also control mechanism recommended by existing standards would therefore be valuable in ensuring that no existing best practices were overlooked in security design

exercises. It would also serve to consolidate subject matter expert conception of how security measures contribute to overall system security goals, such as mission assurance. This research is intended to result in a model-based tool to be used by systems engineers to (i) understand and (ii) design systems security.

Some variant on the Definition modules currently described in Appendix A should be repeated every 2-3 years in order to inform standards with results of other research, and to maintain consistency with new releases of source standard documents.

4.2. Frameworks

All modules in this section deal with security from the point of view of a unique operations environment. Examples of operating environments that may be considered frameworks are command and control systems, unmanned aerial vehicles, and cyberspace networks. They are systems whose security profiles are similar enough for architectural solutions to be useful across ownership boundaries. Security requirements are mission-dependent, and some missions depend more explicitly on security than others. *Framework* in systems engineering refers to a process and related artifacts, such as tools and procedures that are used to arrive at requirements or design. In that sense, the sections that follow describe a framework to be adopted by the researchers, that is, they will emulate systems engineers embedded in an environment that has its own special way of considering security. Note that this section contains examples of such frameworks and systems security engineer research should not be expected to be limited to these. List of potential attributes of security frameworks are readily available in the security literature [21].

These modules have no dependencies because they are meant to take a fresh look at the environment in which commanders and agents operate in a given environment, to learn from those experienced in those environments where security fits in, and what the trade spaces are with respect to security within the given framework. The coordination module will closely monitor these studies, and even preliminary results will be fed into the definition and metrics modules as soon as they become available.

There are an increasing number of opportunities for the systems engineering community to increase its contributions to the enhancement of protection of systems via identifying security solutions that, though identified in the context of a given framework, are extensible to similar situations. In order for these framework studies to achieve these contributions, it will be necessary to:

- identify classes of new reusable system security solutions
- provide a security architectural formulation based on reuse of these solutions
- identify companion security metrics that accompany each new solution and serve to stimulate critical solution design trade-off analyses as part of reuse considerations

These research modules will follow a methodology that supports each of the needs cited above. Researchers will first identify representative classes of what would be new reusable system security

solutions that could be closely coupled to the specific designs and risks associated with the systems that they are intended to protect. Second, they will provide a system security architectural formulation based on reusing security solution design patterns as the potential basis for a continuously expanding set of standard system security architectures for application by the system engineering community. Third, they will introduce an approach to system security metrics based upon the security solution design patterns and the specific risks that they are intended to reduced. Through the coordination module, these metrics are expected to inform the metrics thread of this roadmap.

Though not a necessary criteria for success, these studies would benefit from the assistance of actual systems engineering teams who are engaged in development of systems exhibiting the characteristics of the module's research. For example, a research module may propose to embed an experienced system security research team into an actual on-going project, but limit their role to that of providing hypothetical system security guidance in the various forums of acquisition management, and conducting an analysis of the impact that their recommendations would have had on cost and increased system security capability. In addition, the system security researchers may suggest new documentation requirements, new design review techniques related to system security as they see fit, and provide an assessment of the cost and value that would be achieved with these added activities, based on the system being evaluated. Where this approach is taken, researchers would require access to project teams' requirements and design materials, as well as access to discuss trade space choices with the systems engineers engaged on the project. Researchers may suggest alternative security approaches and make a case for their inclusion in the actual project. Project systems engineers will be at liberty to take or leave the advice, yet researchers would be expected to trace end-state security functionality to trade-space decisions, or other process successes or failures in the project. It is expected that intense concentration on several projects of similar system functionality will make security patterns and features more visible to researchers and thus provide input to the architecture and metrics modules.

This module will study the SSE process applied to different systems operating environments across multiple cooperating enterprise entities. Considerations of the security relationships that relate to the system of-system configuration would be explored, such as alerts to various command center regarding sensor security status, and failure modes of operation that offer resilience, such as sending modified versions of information at lower data rates or through alternative routing in cases where the communication network is disrupted.

This research will also focus on metrics that support the trade space in areas likely to be affected by shared enterprise communication strategies. The include availability and reliability, confidentiality and speed, integrity and completeness.

4.3. Metrics

Security requirements are rarely stated as a set of measured capabilities. An example of such a statement would be, "A unique attribute of a user identity shall be captured and stored by the system. This attribute shall be capable of being recaptured within X seconds, compared to the original capture in

Y seconds, and if such comparison yields a match to within a statistical boundary of 99.9%, an encrypted tunnel (using encryption as per NIST standard) shall be established within Z seconds, and X + Y + Z shall be less than 30; the tunnel shall be capable of sending or receiving a TCP/IP packed with a 2K data payload in W seconds without sacrificing message encryption strength." This type of requirement would be described as *authenticity*. Where such features can be specified in terms of measurable requirements, they map to architectural patterns of the type described in the section on frameworks, and they also yield security metrics. The security metrics will be the subset of the security capability requirements that indicate attainment of the security objectives that drove the requirements setting process. Where security goals are focused only on objectives like confidentiality, the security metrics for the system in this example would include only the user identity management, the encryption strength of the tunnel and packet data payload. Where usability is a key consideration in achieving security goals, the security feature will not be determined to have been attained unless the X+Y+Z < 30 requirement is also met.

Note also that the architectural pattern in this example is that of an information system employing the TCP/IP protocol. The vast majority of research in security metrics has focused on such information assurance or cyberspace concepts of operations. We do not currently have a similar construct for discussing system security metrics in the generic sense. The classic hard problem of "how much security is enough" is difficult even more difficult when the threat landscape changes with systems mission, because there is no way to value a particular security feature or for comparing features. The research proposed in this roadmap will seek to determine what metrics are appropriate, given particular security objectives, by:

- identifying classes of new reusable system security solutions, as described in the previous section on frameworks, and either identify or provide a security architectural formulation based on reuse of these solutions
- identifying companion security metrics that accompany each new solution and serve to stimulate critical solution design trade-off analyses as part of reuse considerations
- encompassing both external threat deflection and internal trust assurance.

Research modules in this section propose a new way of approaching security metrics that involve evaluating metrics in the context of the security framework and definition corresponding to the system of interest. The modules are focused on the identification and measurement of features that strengthen overall system objectives. In this approach, the architecture pattern is used to suggest security metrics [22]. The metrics are performance parameters corresponding to additional features that could be incorporated into the base architecture pattern in order to more effectively thwart potential threats.

4.4. Workforce

Workforce was a major concentration of pre-workshop study. Improvements in workforce training were identified as important related to an understanding a system's security requirements, security relevant aspects of the architecture, security technologies, and understanding the security concept of operations of the system. The skill sets of those in all parts of system design, development, and operation need to be improved with respect to these topics.

Typically, when systems engineers approach hard problems in specialty areas, they seek the advice of an expert. For example, if a system had a requirement to transport acidic material, a systems engineer would seek the advice of a chemist. The situation is the same with systems security. It is common for systems security engineers to supplement systems engineering teams in order to lend security expertise. Research module 4.4.2 on Security Process is an examination of one such teaming strategy, but the examination of the utility of SSE team efforts should not be limited to this one. The integration of security subject matter expertise into the systems engineering process should be a major focus of study in this area.

It was observed that systems engineers and security practitioners cannot be the sole source of security knowledge. We need to improve security cognizance in a variety of segments of the workforce. There is research required to ascertain which training dimensions are required for each workforce segment, and also the extent to which various workforce segments should be engaged in creating training material. Table 1 suggests some workforce training criteria as an example of the objects of study.

Table 1: Workforce Training Criteria

Segment	Training Dimension	Training Contribution
Auditing	System and Security Requirements	Feedback on results
Business	Requirements	System Requirements
Contracting/Purchasing	Requirements	Requirements for Workforce Training
Industrial Property	Requirements	Workforce training
Life Cycle Logistics	Security Requirements Security Technologies Concepts of Operation	Workforce Training Skill Sets
Program Management	System and Security Requirements Security Technologies	Workforce training System and Security Requirements
PQM (production, quality, and manufacturing)	System and Security Requirements Models and Tests for Security Verification and Validation Security Technologies Concepts of Operation	System and Security Requirements, including verification and validation Concepts of Operation

Table 1: Workforce Training Criteria		
Segment	**Training Dimension**	**Training Contribution**
SPRDE (Systems Planning, Research Development, and Engineering)	System and Security Requirements Models and Tests for Security Verification and Validation Security Technologies Concepts of Operation System Architecture	System and Security Requirements, including verification and validation Concepts of Operation
Test & Evaluation	System and Security Requirements Concepts of Operation Security Verification and Validation	Test Criteria for System and Security Requirements
Requirements Management	System and Security Requirements Security Technologies Concepts of Operation	System and Security Requirements

Once appropriate training requirements have been established, then tools should be created to accomplish the training. Preparation of the curriculum will require one or more pilots of proposed curriculum, and an evaluation of the pilot's effect on systems security resulting from the workforce activity. It is expected that the pilot organization be involved early in the process in order to perform pre-education-delivery studies that would be necessary to produce statistics demonstrating curriculum effectiveness.

4.5. Systems Engineering Methods, Processes and Tools

This section addresses problem statements related to the development and exploitation of security knowledge bases, asset libraries, and specialist communities. Research on metrics and frameworks is expected to yield new models and tools for SSE. In order to move these from the research stage to practical use, there must be some verification and validation processes that ensure security metrics demonstrate that security requirements are met. Once established, these verification and validation processes themselves must be verified and validated to accomplish SSE goals. This type of activity could proceed via publications and pilots. It could be supplemented with surveys and other SSE community feedback mechanisms.

There are many MPTs that are well established in disciplines that are related to security or have similar goals or objectives. Successful MPTs in these areas should be examined for possible application to systems security. If the MPTs in the toolset of nearby disciplines seem to be applicable to system security, this could provide a quick and easy method of expanding the toolset of metrics currently available to SSE. Without initially focusing on any one tool, a research team could identify and analyze related disciplines like Safety, Reliability, and Surety in search of MPTs that have relevance in system security engineering. These tools would then be prioritized for further study as to their utility in the security space. They would be added to the potential set of tools to be included in related research in frameworks and architecture metrics.

All MPTs resulting from research in the roadmap are expected to be made available to systems engineers via a section devoted to security in the BKCASE standard. Drawing on results from other tasks, MPTs would contribute systems security knowledge to BKCASE. The ideal outcome is that BKCASE will be supported worldwide by the Systems Engineering community as the authoritative body of knowledge for the SE discipline and that the GRCSE will receive the same global recognition and serve as the authoritative guidance for graduate degree programs in SE. If the results of the security standards and process work can be incorporated into BKCASE, we will thereby leverage the extent to which the BKCASE process has achieved global outreach in the service of systems security.

4.6. Advanced Research Topics

Systems thinking words to describe features to anticipate unknowable and therefore unexpected interaction with other systems are *agility* and *adaptability*. These describe a system that contains inherent non-equilibrium and processes information from internal and external sources as feedback to enable change and growth. This is best understood in contrast to a system that simply processes information in predefined ways to achieve specific predetermined outcomes. Such systems are frequently described as self-organizing, as they possess ability to reorganize their internal state, interfaces, or other functional components in response to new information [8]. This suggests that a system's level of agility and adaptability may contribute to its response-ability to the changing operational environment and threats.

Because self-organizing systems can structure or restructure themselves as needed to respond to external information, the exact shape, definition and behavior of a self-organizing system becomes an ongoing, interactive adaptation to the conditions of its environment or situation. This has implications for system architecture—particularly if threats are self-organizing systems and the architecture sets up the conditions for systemic security to emerge from a system that may or may not include self-organizing security features.

Security events tend to disrupt system operations because systems, and the assets they contain, are adversely impacted by such events. Increased levels of systemic security include not just ways to deflect attacks, but also ways to respond to security events that minimize their harmful effects. Some level of disruption may persist for weeks or months and will occur regardless of whether the precipitating event is a natural disaster, an accident, or an attack. While security program management and risk mitigation planning may operate under different premises when considering these different types of events, all require a combination of prevention, detection and response capabilities. To maintain stakeholder confidence in the system and to minimize an event's harmful impact, the system must respond effectively regardless of the cause of the event.

A system's ability to respond to threats is a function of its design and architecture; the effectiveness of its control information collection, analysis and risk-based decision making; and the system's responsiveness to those risk-based decisions. Systemic security metrics provide the capability to

translate data into useful information to support decisions with respect to system features as well as its human response capability in the context of an operational environment.

These modules are expected to explore the degree to which internal components of a system enable—or may be redesigned to improve—response-ability and emergent systemic security. System security engineering is concerned with how the architecture might be designed, using self-organizing capabilities or controls, to improve the systems' overall security via the self-organizing behaviors of the system in its operational environment.

4.7. Coordination

Coordination will be required to provide oversight adequate to ensure that research results transfer between modules, especially as new researchers join these endeavors. It is also expected to be the source of information sharing opportunities both within the SERC community, such as the workshop in which SERC researchers shared their experiences in SSE, and outside of the SERC community, such as a SERC/INCOSE working relationship. The theme overall is to rely on the coordination modules to provide the leadership to integrate the best results from all modules into new standards and educational materials going forward.

This coordination module would also bear the burden of monitoring progress on this roadmap, modifying it to accommodate new developments in the field of security, and specifying new requirements for future systems security research.

5. Summary and Next Steps

This research roadmap reflects its vision for success in that it sharply focuses on how to equip systems engineers with the MPTs to recognize and specify criteria for mission assurance. It is designed to identify and illustrate systemic and repeatable security controls, as well as quantify their effectiveness. Research modules recommended in this roadmap are all related in that they serve to enrich the security-related endeavors of the systems engineering community. However, they purposefully do not replace the systems engineering community activities. In recognition of the breadth of the systems engineering field, this research program does not propose to build any single framework, assurance architecture, or reference model. Rather, it will provide multiple reference models with properties that map to frameworks that may or may not exist today, but are intended to inform trade-space decisions with respect to security both now and in the future.

Although all the potential research tasks described in Appendix A would provide value to the SSE, some ideas have already been singled out for further research. These are:

- Model-based Security Standards Compliance
- Nearby Disciplines
- Framework Metaphorical Modeling

- Architectural Security Metrics
- Systems Engineering Security Workforce Education

It is hoped that an early and sharp focus on existing systems security standards from a systems modeling viewpoint will highlight commonality among standards as well as be of assistance to other research tasks that will attempt to identify specific system attributes and mechanisms that increase systems security. Common models of security standards will help align research in the context of the overall roadmap, and provide individual researchers with the ability to easily identify what the benefit of their research will be to the overall security landscape. Note there is no assumption here that following standards will increase security, but simply that understanding what security professionals have spent decades achieving will be of value to future security work. Moreover, as systems engineers are familiar with models and also are increasingly required to comply with this, this work should result in a systems engineering job aid.

An early and sharp focus on nearby disciplines is also expected to provide tools of value to the systems security engineering community. Safety, reliability, resiliency, and control systems endeavors have a longer history of focus for the systems engineering community. Methods, processes, and tools that have emerged from these endeavors will allow. As roadmap contributor Dan Geer remarked [as quoted in 23], "Civil engineers know why bridges fall down, lawyers know the difference between policy and enforcement, doctors know the terrible demands of making life-and-death decisions under uncertainty, public health practitioners know that the great triumphs over disease began with sewers not with antibiotics, preachers know that great thoughts cannot be transmitted without the vehicle of familiar tales in which to embed the higher principles, and on and on … we must spend it with as much wisdom and perspicacity and dedication as we can muster."

Following the same line of reasoning that leads to the prioritization of nearby disciplines, there is a consensus on the perceived value in close scrutiny of security architecture frameworks. Systems of different types have completely different security requirements and profiles. Yet there is a lack of recognition that systems security requirements may be unique. Concentration on evaluation of security measures with respect to a given framework will heighten recognition that not all security systems engineering must be tightly coupled with system context and functional decomposition. This approach is expected to lead directly to architectural security metrics. The results of all of these activities are intended to improve the security effectiveness of the systems engineering workforce. We include this item in the next steps list in order to ensure that all other activities maintain this focus.

6. Contributors

Ideas included in this paper were collected from a variety of sources, primarily the SERC Security Engineering Workshop held in Washington D.C. on March 31- April 1, 2010.[2] The first four scope

[2] Appendix C is the program from that workshop.

categories had been methodically explored via problem definition statement, literature surveys, solution criteria, and next steps by the workshop program committee in advance of that workshop. These ideas were socialized with the SERC community and selected members of the security systems research community outside of the SERC. Solution proposals were discussed at the SERC Security Engineering Workshop. Workshop attendees and invited reviewers shared research ideas both at the workshop and in the form of post-workshop conference notes, recommendations, and advice. The research ideas have been synthesized into the set of interrelated modules. This roadmap also contains contributions from those who reviewed a preliminary draft from both inside and outside the SERC, including the Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University, who joined the SERC Security Systems Engineering Team after the workshop had been completed.

Members of the Workshop Program Committee and primary authors of this report are:

Jennifer Bayuk, Stevens Institute of Technology
Dennis Barnabe, NSA/ESEA
Jonathan Goodnight, OUSD(AT&L)/DDRE/SE
Drew Hamilton, Auburn University
Barry Horowitz, University of Virginia
Clifford Neuman, University of Southern California
Stas Tarchalski, Stevens Institute of Technology

Workshop Attendees and contributors include:

Daniel E. Arista, Syracuse Research Corporation
Michael Atighetchi, Raytheon BBN Technologies
Kristen Baldwin, OUSD(AT&L)/DDRE/SE
Peter A. Beling, University of Virginia
Barry Boehm, University of Southern California
Timothy Busch, Air Force Research Laboratory Information Directorate
Lori A. Clarke, University of Massachusetts
John F. Clem, Sandia National Laboratories
Germain Creamer, Stevens Institute of Technology
Paul Croll, Computer Sciences Corporation
Rick Dove, Paradigm Shift International
Sonja Dua, Stevens Institute of Technology
Robert Edson, Analytic Services Inc.
Lieutenant General Robert J. Elder Jr, USAF Ret.
Jeremy Epstein, SRI International
Glenn Fiedelholtz, Department of Homeland Security
Richard Hale, Defense Information Systems Agency
Joe Jarzombek, Department of Homeland Security, National Cyber Security Division
Georganne B. John, Analytic Services Inc.
Khaldoun Khashanah, Stevens Institute of Technology
Linda Laird, Stevens Institute of Technology
Gregory N Larsen, Institute for Defense Analyses
Lucas Layman, Fraunhofer Center at UMD
Mark Lorenc, Los Alamos National Laboratory

Barbara Maguschak, The Aerospace Corporation
Chad Manifold, Stevens Institute of Technology
Richard Marshall, Department of Homeland Security, National Cyber Security Division
Archibald McKinlay, Dept of Navy, ASN(RD&A)CHSENG
Dr. John F. Miller, The MITRE Corporation
Joe Mitola, Stevens Institute of Technology
Louis Montella, The MITRE Corporation
Ali Mostashari, Stevens Institute of Technology
Janet Carrier Oren, Stevens Institute of Technology
Leon Osterweil, University of Massachusetts
Paul Popick, Aerospace
William Pryor, NGA
Jose Ramirez-Marquez, Stevens Institute of Technology
Kenneth Reese, Space Dynamics Laboratory
Paul Rohmeyer, Stevens Institute of Technology
Tony Sager, NSA
Raghu Sangwan, Pennsylvania State University
Ken Shotting, Department Of Defense
Forrest Shull, Fraunhofer Center at UMD
Justin L. Smith, NGA
Russell Cameron Thomas, Meritology
Carla Ulibarri, Sandia National Laboratories
Carol Woody, Carnegie Mellon University
The Honorable Michael W. Wynne, USAF Ret., and Chair, SERC Advisory Board

Other reviewers and contributors include:

Julia Allen, CMU
Suhair Amer, Southeast Missouri State University
Dekle Charles, OSD
Fred Cohen, California Science Institute
Judith Dahmann, OSD
Dan Geer, InQTel
Anup Ghosh, GMU
Jo Ann Grout, MITRE
Rebecca Horton, Sandia National Labs
Havlicek Jeff, USAF
Scott Lucero, OSD
William Martin, National Security Agency
Michael May, ATL/DRE
Peter Neumann, SRI
Donn Parker, retired cybersecurity luminary
Greg Shannon, SEI CERT
Dan Schutzer, FSTC
Ted Serbinski, Navy
Gene Spafford, CERIAS, Purdue

We also acknowledge the contributions from the efforts performed under federal contract number: FA8240-07-C-0141-P00004.

Appendix A: Additional Detail on Selected Research Modules

This Appendix includes research modules suggested by workshop participants and reviewers as being germane to the goals of the roadmap elements as described in Section 4. As noted in Section4, actual research tasks inspired by this roadmap may of course contain other elements of interest to sponsors that would affect scope as well as expected timeframe. The list of modules included in this appendix is:

Security definition (Reference Section: 4.1)
- A. Security Standards Reconciliation
- B. The Utility of Security Best Practices
- C. Security Policy Compliance
- D. Adaptation of Security Policy and Mechanism

Security Frameworks (Reference Section: 4.2)
- E. Critical Program Information Protection
- F. System of Systems
- G. Configuration Hopping
- H. Continuity of Communications
- I. Data Continuity Checking
- J. Denial and Deception
- K. Shared Command Information Sharing
- L. Physical Security Frameworks

Security metrics (Reference Section: 4.3)
- M. Architecture Metrics
- N. Risk Metrics
- O. Security versus Convenience
- P. Security Trade Spaces in Emerging Technologies
- Q. Trust Assessment Models

Security workforce (Reference Section: 4.4)
- R. Workforce Education
- S. Security Requirements Process
- T. SE Career Path

Security MPTs (Reference Section: 4.5)
- W. Exploring Nearby Disciplines
- X. BKCASE Security Section

Security advanced topics (Reference Section: 4.6)
- Y. Agile Architecture
- Z. Executable Architecture
- AA. Critical Functionality

Security Research Coordination (Reference Section: 4.7)
- BB. Coordination
- CC. Hypothesis Test

Security definition (Reference Section: 4.1)

A. Security Standards Reconciliation

Security Standards Reconciliation	
Problem Statement	Systems designs and operating environments are increasingly required to comply with a wide variety of complex security standards documents. Currently, systems engineers independently reconcile such standards to system requirements on a project by project basis, resulting in countless hours of rework without reuse. The complexity renders the current security requirements process not only inefficient but error-prone.
Background	Specific recommendations included in security standards are all based on best practices in securing complex systems environments, and so are repeated in multiple instances of them. For example, a National Institute of Standards and Technology (NIST) publication on security metrics describes itself as a recommended methodology for complying with requirements in a companion self assessment standard which references 11 other best practices documents as sources for control [24, 25]. Another example is the US NIST's Recommended Security Controls for Federal Information Systems, which also refers to several other standards documents [26]. Systematic application of these standards has become synonymous with due diligence in establishing system security.
Solution Criteria	A unified model for security standards that is comprehensible to systems engineers and easily applied to a wide variety of systems.
Next Steps	The research would be expected to catalogue and model 15-20 influential security standards into a common systems engineering modeling tool. It would produce a guide for systems engineers to use to ensure that they are compliant with one or more specific standards.
Thread	Definition
Dependencies	none
Timeframe	Short-term

This module will identify common elements in existing security standards used as checklists in current systems engineering processes. These include the Systems Security Engineering Handbook, the Joint Software Systems Safety Engineering Handbook, NIST and ISO security management standards as well as functionality standards such as those set by common criteria and compositional standards such as the building security in software security standard. Members of this research group will monitor the Open Group product security standards activities and drafts. Each of these standards would be modeled using systems engineering modeling techniques designed to highlight similar concepts and relationships between recommended activities and systems configurations.

In order to clarify the security recommendations included in a given security standard, the standards documents will be modeled using a systemigram [11]. The word systemigram was coined by as a convergence of "system" and "diagram." It was envisioned as a tool to assist systems engineers in covering a topic without sacrificing detail required to accomplish clarity. A systemigram starts with the system to be defined, and includes nodes and links. Nodes are nouns. Links are verbs. A systemigram is read by focusing on a noun that is part of a system and following the links from it, reading the verbs to understand the relationships between system components.

Using the systemigram to model a security standard, we can test its ability to measure system security. The result is that systems security is identified orthogonally. By comparing standards to each other, as well as to the systems security features that provide system assurance, it will be possible to demonstrate that the extent to which a security standard provides value to a system security posture.

This study is important precisely because it has never been done. Security standards to date have been composed by consensus based on examples of organizations that have compiled security controls in response to known threats. The compositional approach has widespread adoption due to industry consensus rather than due to any attempts at academic justification. This study will look holistically at the set of controls that has been compiled into a standard. It will also look holistically at system goals for security and identify gaps where standards do not address those goals. Because the systemigram modeling approach incorporates concepts such as hierarchy, boundaries, and emergence, this study will also undoubtedly identify patterns of high level security features that are similar across systems in different domains.

B. The Utility of Security Best Practices

The Utility of Security Best Practices	
Problem Statement	Although security standards and best practices have been accumulating for decades, malicious activity in cyberspace is not thwarted simply by application of those standards.
Background	Today's literature cannot adequately answer research questions with respect to security metrics. Nevertheless, consistent application of cybersecurity standards using expert security risk judgment have been refined and adopted over the years, and numerous publications show consensus among security experts that this does increase the overall cybersecurity level of the target system [27]. However, there actually have not been formal studies that prove whether or not such diligence in security configuration does increase system security.
Solution Criteria	A unified model for security standards that is comprehensible to systems engineers and easily applied to a wide variety of systems.
Next Steps	The research would be expected to utilize the model provided by research module A, and to devise scientific studies to determine whether it as a whole, or parts of it, provide security value.
Thread	Definition

Dependencies	Reconciliation of Security Standards
Timeframe	Mid-term

In the executive summary of this document, we acknowledge that, although security standards and best practices have been accumulating for decades, malicious activity in cyberspace is not thwarted simply by application of those standards. Rather, cyber-perpetrators utilize the same cyberspace services that are available to those who are authorized to use them. The goal of a cyber-intruder is rarely to damage a system, but to exploit it to gain objects of value. Cyber-incidents of espionage and fraud are more common than cyber-terror. Cyber intruders study our security standards in order to avoid them as they move seamlessly through our systems masquerading as authorized users.

Hence compliance with security standards is not an adequate metrics by which to judge whether a system is secure. This study placed confidence in the ability of soft systems engineering methodology to support a structured approach to the determination of whether a system may be considered to be secure in the context of its mission or purpose. Moreover, current methods of collecting security metrics may be useful in the process. The knowledge gap lies in the ability of the field of security metrics to properly assess whether those controls were appropriately selected, given system security requirements. This research will fill that gap by strengthening current capability to assess security with respect to requirement. Groundwork had been laid for using systems thinking as an approach to security architecture issues. Recent work by Wirsbinski proposes using systems thinking concepts as a method for improving the quality of security assessments [28]. This research will extend these efforts into an systematic approach that measures whether a system design that embeds a security solution meets security requirements.

C. Security Policy Compliance

Security Policy Compliance	
Problem Statement	Systems engineers that are not security specialists do not easily match systems security requirements to policy specifications.
Background	Current security systems use complex and fine grained policy sets that require configuration by security specialists who understand the technologies but not the system as a whole. This creates a disconnect between the real goals of a system, and the security policies implemented by the system – often creating systems that are difficult to use or which do not focus on the importance security aspects of the system (as opposed to merely checklists).

Solution Criteria	Through the use of tools to capture the high-level, often coarse-grained, information and control flow constraints of a system, the high-level system focused intent of security policy can be enforced at the network, operating system, and middleware components of a system in a way that augments the finer grained policies that typically require configuration by security specialists. A failure of fine-grained controls would thus still be contained through the high-level policy compliance standards established during system specification and design.
Next Steps	Development of methodology and language to be employed in specifying system requirements for information and control flow and resilience. Recommendation and testing of changes to security mechanisms to utilize this new type of security specification.
Thread	Definition, Workforce, MPTs
Dependencies	none
Timeframe	Mid term

This module uses the word security policy to refer to technical configuration specifications designed to achieve a security goal. In acknowledgement that security goals can be met using alternative technical specifications, it will attempt to demonstrate compliance with higher level goals using reference to known capabilities provided by existing configurations. It would build on advances in configurations designed for policy compliance such as SCAP.

D. Adaptation of Security Policy and Mechanism

Adaptation of Security Policy and Mechanism	
Problem Statement	High-level security policies to be enforced by a system may change over time, but security features tend to be too brittle to meet this challenge.
Background	Over time, how systems are used changes. Changes, such as the environment within which the system operates, may drive changes to high- level information and control flow and resiliency policies. There needs to be a way to capture this high level change in system requirements, and ensure that the security specific mechanisms implemented within the system are able to operate within the new environment, or that problems doing so are identified to individuals at the appropriate level of responsibility.

Solution Criteria	Through the use of tools to capture the high-level, often coarse-grained, information and control flow constraints of a system, the ability of the network, operating system, and middleware components to meet those goals must be assessed, and the new policy enforced. If the high level goals cannot be met by the existing system, an exception must be raised to individuals with an appropriate level of responsibility. This process becomes part of the accreditation of a system to operate in new environments.
Next Steps	Development of methodology and language to be employed in specifying flexibility in system requirements for information control flow and resilience. Methods to assess the ability of a system to meet the specified requirements.
Thread	Definition, Workforce, MPTs
Dependencies	Security Standards Reconciliation, Security Policy Compliance
Timeframe	Mid term

This module relies on the ability to represent security policy as dynamic requirements that change relative to framework and requirements. It would benefit from architectural interpretations of security standards and policy, and so is reliant on progress in those areas.

Security Frameworks (Reference Section: 4.2)

E. Critical Program Information Protection

Critical Program Information Protection	
Problem Statement	The impacts of managing requirement changes frequently disrupt CPI protection profiles and also introduce vulnerability to previously unexpected emergent security threats. There is no specific program to prevent these types of events from occurring without detection while operating within the normal DOD acquisition management system.
Background	It is DoD policy to provide uncompromised and secure military systems by performing comprehensive protection of CPI through the integrated and synchronized application defensive countermeasures to mitigate risk. It is also policy to extend the operational effectiveness of military systems through application of appropriate risk management strategies, employ the most effective protection measures, to include system assurance and anti-tamper, and document the measures in a Program Protection Plan.
Solution Criteria	An acquisition-experienced system security team will be expected to observe change control cycles within an existing DoD or similarly complex enterprise, recommend appropriate security engineering impact analysis on CPI, and conduct an analysis of the impact that their recommendations would have had on cost, other trade-space factors, and increased system security capability.

Next Steps	The first several months of this effort would involve defining the future activity in detail, selecting the appropriate system security acquisition management team, and the program(s) to be used for evaluation. Following, the research would involve conducting the actual analysis and evaluation.
Thread	Frameworks
Dependencies	none
Timeframe	Long term

There is a criticality working group within the DoD whose focus is engineering for criticality. This research module would provide support in the form of knowledge engineering and information classification and collection of knowledge types as they relate to critical mission assurance. Individual, explicit, social, declarative, and procedural knowledge would be sought from subject matter experts engaged in critical mission engineering efforts. Knowledge storing and sharing techniques would be used to ensure that security of mission assurance could be well articulated early in the requirements process in ways that could not be traded out while considering design alternatives.

F. System of Systems

System of Systems	
Problem Statement	The life cycle management of security across the enterprise, including synchronizing interdependent changes to security, is greatly simplified by considering security as a separate system with a well-defined interface. Yet, as the likelihood of being the victim of a successful cyber attack continues to grow, the system-of-systems configuration provides opportunities for increasing resilience to attacks.
Background	The rapidly increasing extent to which systems are integrated via service oriented architecture requirements has resulted in systems operating models that were never envisioned by architects and engineers. Rather, these system integration capabilities are often solely software-enabled and resulting systems interfaces between trusted systems operators are established without a security risk assessment of the newly conjoined system of systems.
Solution Criteria	The systems engineering community could be developing solutions that on the one hand create added security by exploiting resources available at the entire system-of-systems configuration level while maintaining the convenience of security solutions that reflect the current practical considerations of management, including interoperability, reliability, availability, and maintainability.
Next Steps	Analysis of examples of systems of systems configurations to identify architectural patterns and associated impact on security. Identification of a requirements set that would motivate a more secure design, as well as supporting security features at both the individual system and system-of-system level.

Thread	Frameworks
Dependencies	None
Timeframe	Long term

Over time, large enterprises sequentially develop new systems and modernize existing systems that support various functions or various organizations. It is a common occurrence that these systems are integrated to achieve greater value, resulting in a system-of-systems. While the overall system-of-systems serves the enterprise, frequently each system lifecycle is managed and operated by a particular organization within the enterprise. The purpose of this research is to identify security features at both the individual and interface level that address systemic security threats and vulnerabilities due to composability issues.

For example, design patterns can be developed for peer systems in an overall system-of-systems configuration, wherein one peer provides back-up services to another in the event of a successful denial of service attack. These back-up capabilities can either be fully redundant, or alternatively can offer partial coverage for the functions that need to be replaced. Similarly, design patterns can be developed where peer systems are used to help isolate the existence of a difficult-to-detect attack that manipulates or steals data. This can be accomplished, for example, through data parsing and continuity checking wherever data crosses the boundaries of an individual system and serves one of its peers. Design patterns such as these provide a starting point for exploring the flexibility of the practical management constraints that limit system-of-system solutions, so that over time the systems engineering community can establish a generally accepted understanding of what is deemed as acceptable from a management point of view, and what is not.

With regard to metrics, it is likely that many of the design patterns would be derivatives of related patterns used for a single system. For example, the system-of-systems security backup pattern may be considered an extension of the physical configuration-hopping pattern in that, in both cases the configuration of the system in need would be supported by a peer system in addition to its own processing. However in the single system case the redundancy must be non-interfering, whereas in the system-of-systems case it may be acceptable to suffer some losses in performance. Metrics that would be useful in judging the security level of such a solution would therefore also include the number of operating system platforms that a peer could hop onto, the time it takes to accomplish a hop, the extent to which the hopping system can automatically reconfigure its named interfaces, and the loss in performance of the backed-up system when operating in the design pattern's failure mode. In addition, it would also include metrics related to the back-up host peer, such as the amount of processing power it is able to lend to a peer without significant performance degradation, and the extent to which its own performance would be impacted.

The system-of-systems security data leakage pattern metrics may be supported by measuring the number of data types that are included in parsing, the number of protocols that are covered by the data content inspection feature, and the percentage of network and local system interfaces that are covered.

It may also be extended to reporting and alerting mechanisms to be used in cases of identified data leakage.

G. Configuration Hopping

Configuration Hopping	
Problem Statement	The threat of concern is a Trojan horse embedded in a critical software component that can be used to significantly impact system operation. Added security would be provided by dynamically hopping to different versions of the selected software components operating under different operational configurations.
Background	This requirement addresses operational integrity by focusing on a selected set of software components that are considered by system designers as "critical" to proper system operation. The variations can be achieved through dynamic switching of virtual machines as well as through switching of physical configurations. A similar design pattern could be developed for malicious hardware components as well.
Solution Criteria	A successful result of this research module would produce a working prototype wherein software components required for mission assurance could hop across platforms without impact to system operations.
Next Steps	Identification of a framework that relies on software modules wherein interprocess communication requirements are well understood and formally modeled. Reproduction of that software in a lab environment where recompilation, redesign of communication interfaces, and multiple platforms that could feasibly be made available in the framework environment.
Thread	Frameworks
Dependencies	none
Timeframe	Long term

This security design pattern addresses operational integrity by focusing on a selected set of software components that are considered by system designers as "critical" to proper system operation. The threat of concern is a Trojan horse embedded in a critical software component that can be used to significantly impact system operation. Added security would be provided by dynamically hopping to different versions of the selected software components operating under different operational configurations. The variations can be achieved through dynamic switching of virtual machines as well as through switching of physical configurations. A similar design pattern could be developed for malicious hardware components as well.

Consider the example of a financial institution's private Cloud computing configuration supporting a real time information system used for decision support for stock trading. In order to provide shared computing services to support all of the financial institution's computing needs, Cloud computing

architectures include configuration control capabilities that can be used to support varying user demands in terms of software infrastructure requirements (operating systems, network interfaces, etc) within the shared Cloud infrastructure. Two basic technical features of Cloud infrastructures are the use of virtual computing controlled by a "hypervisor" as a means for managing multiple operating systems running on a common hardware base, and use of physical configuration switching in the event of an outage of a Cloud facility. These already existing Cloud capabilities can be used to provide users with security related controls that are based upon computer configuration-hopping (both real and virtual). The configuration variations would provide security by reducing assurance to attackers about which applications will be running on which virtual machine or in what physical configuration at any given time. Selection of hop rates would be tied to system related properties of the application being addressed. The application owner would perform a risk analysis at the application layer level to determine which specific services to replicate on which virtual machines, and which services to have executed on which physical part of the Cloud infrastructure on a time varying basis. The selection of components to hop would be based on the decisions being supported and the consequences of wrong decisions. Hopping rates would need to account for the dynamics of decision making. Physical switching can also be used to address the potential of an insider threat by switching system administrators as a bi-product of the physical switching process. The Cloud service provider would need to sustain critical application performance requirements, accounting for any performance degradations due to switching, and would provide users with a cost for sustaining performance objectives in the face of switching.

The example presented above highlights the close relationship between the security solution and the system that it is securing. The selection of software components to hop, the number of replications to develop for hopping purposes, the specific operating systems to hop across, and the specific design parameters for hopping, all require intimate knowledge about the system being secured and correspondingly make the system engineers the logical source of the solution. As in the case of continuity checking, the reusability of a software hopping function would be of interest, and a design pattern could be established for reuse across a broad set of systems.

To measure the configuration hopping architecture using an architectural security metric, we would measure attributes of the design that increased or decreased its ability to accomplish its security goals. Measurements that would be relevant to the configuration hopping security feature might include, but of course would not be limited to, the number of platforms that an image could hop among, the time it takes to accomplish a hop, and the extent to which the system can automatically reconfigure its named interfaces.

H. Continuity of Communications

Continuity of Communications	
Problem Statement	Since the dawn of electronic communication, there have been attacks intended to disrupt communications between logically or geographically separated components of a system. This module is concerned with deterring and responding to those attacks.

Background	A well recognized denial of service threat to systems involves physical disruption of communications that connect separated elements in a system. A frequently used design pattern to respond to this kind of threat is the provisioning of redundant communications sources, such as receiving redundant landline communications through alternative routing paths that are spatially separated so as to make acts of sabotage more difficult to carry out without being discovered during the attack.
Solution Criteria	The redundant communication design pattern should be expanded to include multiple modes of communication as well as innovative approaches to data protocols which would enable the addition of completely new communication scenarios.
Next Steps	The full spectrum of currently available communications equipment should be surveyed for applicability to secure communications. Trade-space criteria for various real-world communications scenarios should be established, and technologies identified that would provide new alternatives for secure communications strategies.
Thread	Frameworks
Dependencies	none
Timeframe	Long term

Continuity of communications is often achieved via standard backup methods such as supplementing landline communication with multiple radio communications systems. However, for systems where the normal bandwidth requirements for data transfers exceed the available bandwidth of a radio system used to provide continuity, functional components of the system must be modified to either operate in modes with larger delays in receiving data, or in modes that can acceptably work with reduced data content in order to keep communications delays within the normal system specifications. This security design pattern would provide the system adjustments necessary for either accommodating greater delays or reducing the amount of data transmitted.

For example, communications to support military warning systems are subject to sabotage and electronic warfare. Data delays must be kept to a minimum because warning is a precursor to what can be time-critical responses. As a result, for applying this system security design pattern, data sent over low bandwidth radio systems for providing continuity of long-range communications must be compressed to avoid unwanted delays. As a particular example, a warning system might normally receive relevant remote sensor surveillance information over dedicated landline telecommunications system at 9600 bits per second. For that same system it may be desirable to use an HF radio system as a redundant source, but only capable of delivering the data at 2400 bits per second. The system designer may choose to delay the data by a factor of four, or alternatively the data can be compressed by a factor of four. Assuming that added delay is not acceptable, compression is required. One method of compression could be to change data quantization levels. For example, for a warning system that receives communications regarding locations of sources of attack, instead of sending locations with 0.1

mile precision the communicated data can locate the sources with less precision (e.g., 1 mile precision), thereby reducing the number of bits required for transmissions. Further assume that the system designers would like even greater security regarding disruption of communications and would like to add a lower frequency communications system that supports only 100 bits per second of communication in addition to the HF system. This large a reduction from the normal 9600 bits per second sent by landline could warrant the design of a new mode of system operation that uses summary reports rather than individual location reports (e.g., "there are 3 sources of attack coming from the northern sector of enemy locations") with only the number and sector description being the communicated data. While the specifics of this example would be highly dependent on the system design that was being secured, the general design pattern could likely be used on a variety of systems requiring similar security solutions.

Security metrics corresponding to this design pattern would include the number of physically different communications paths, the number of logically distinct communications protocols, and information-theoretic statistics that demonstrated ability to provide mission critical information when operating at reduced protocol capacity.

I. Data Continuity Checking

Data Continuity Checking	
Problem Statement	Important surveillance system designs integrate pipeline computing processes. These typically start with a data collection function, and progress through a pipeline of data processing including computational processes such as object detection, object location tracking, integration of correlated reports from multiple data collection sources, object identification, and object presentation to operators responsible for managing or responding to observations. Such configurations have common security threats, including on-command Trojan horses that could either: a) prevent data from being properly processed in order to avoid operators from observing a specific object(s); or b) create artificial data as a decoy to attract Operator attention away from other data that should be acted upon.
Background	Examples of systems that include such capabilities are air traffic control systems collecting radar surveillance reports to support air traffic management functions, and military warning systems collecting infra-red and radar reports for alerting our nation's leaders to an Intercontinental Ballistic Missile (ICBM) attack so as to enable timely military responses. In practice, the hardware/software architecture for executing such processes varies, ranging from centralized computing configurations to highly distributed computing configurations.
Solution Criteria	Development of taxonomy for relating data elements to decisions in a manner that helps system users to relate externally forced changes in decision support data to potential critical decision errors.

Next Steps	Identify security solutions tightly coupled to the system design, such as data continuity checking, and create architectural patterns that will strengthen system assurance. For each, design method of measuring relative strength of security attribute such as confidentiality, integrity, and availability.
Thread	Frameworks
Dependencies	none
Timeframe	Mid term

Important growing threats such as hardware and software Trojan horses that can manipulate data presented to the decision support system operators, and insider threats that can command the Trojan horses should be considered in command and control systems design and evaluation efforts. Evaluation techniques that map operator processes into projected damage outcomes when data is improperly manipulated are required be used to decide on the level of data assurance to apply on a specific data item basis, related to the role a data item may play in critical decisions.

In this module, command and control threat scenarios would be developed to provoke design choices from an adversarial viewpoint. Considerations of the security relationships that relate to the system of-system configuration would be explored, such as alerts to the command center regarding sensor security status, and failure modes of operation that offer resilience, such as sending modified versions of sensor information at lower data rates or through alternative routing in cases where the communication network is disrupted. The outcomes of the experiment would be compared to the outcomes that would have resulted from limiting solutions to the traditional set of perimeter security approaches, using current best practices.

After the fact analysis raises the question of why wasn't the warning system designed to recognize a condition where there was data indicating an attack on the screens being observed by operators, while at the same time there was no missile-related data being received from the system's sensors. Such checking is referred to as data continuity checking, and the only answer to this question is that the systems engineering community did not recognize this as a needed security feature for the warning system. While this event was stimulated by malfunctioning hardware, it could just as well have been a supply chain injected Trojan horse (hardware or software) controlled, for example, by an insider.

The design of such a data continuity agent would require:

1. Development of a taxonomy for relating data elements to decisions in a manner that helps system users to relate externally forced changes in decision support data to potential critical decision errors. Examples would include:
    - identification of single data elements that can change a critical decision regarding a single object (e.g., "friend" or "foe" designation)
    - identification of single data elements that can change a decision regarding pairs or groups of objects (e.g., change of the observed altitude for a radar observation of an

aircraft so as to create what appears as a possible collision opportunity with another aircraft)
- elimination of data that the surveillance system is designed to deliver to responders; or
- creation of false data that diverts the efforts of responders.

2. Development of a user interface to the data continuity agent for:
   - designating the meta data that will be available to the agent from the various service components in the system,
   - the comparisons that the agent is required to make as a basis for recognizing a discontinuity,
   - the time lines to be used for making comparisons;
   - and the interface that the agent should use for reporting discontinuities.

These design issues require research into sensor data analysis and operations, such as coordinating the activity of autonomous sensors [29]. The reusability of data continuity checking agents would also be of interest, and it is possible that such agents could be designed for reuse across a defined set of system specifications.

J. Denial and Deception

Denial and Deception	
Problem Statement	Denial and deception have been successfully used by adversaries in gaining strategic advantage in realms from intelligence gathering to politics. Systems that focus on protecting critical program information would thus be missing a strategic opportunity to learn from these adversarial approaches and employ such denial and deception measures in the service of systemic security.
Background	This module is based on the recognition that denial and deception strategies are commonly used by the adversary and could be productively used as a defense strategy.
Solution Criteria	Where honeypot or cloaking techniques are used to lure potential adversaries away from critical program information, such systems would presumably have common characteristics which would inform the architecture and metrics modules of this research program.
Next Steps	This research effort involves the design and implementation of a variety of honeypot architectures, and quantifying the cost-benefit analysis trade-offs of various honeypot features.
Thread	Frameworks
Dependencies	none
Timeframe	Long term

Denial and deception is a tools often used by adversaries but not explicitly part of a security program. However, growing recognition that this tool is skillfully employed make it possible for a security program

to some extent rely on its existence. This may facilitate security goals by allowing advantage to be taken of unauthorized observation of security program elements.

The security feature here involves detecting and exploiting unauthorized access, which necessarily includes a concept of authorized, so it is distinguishing between authorized and unauthorized is critical to the success of this effort. This is often accomplished by setting up a place where everything is unauthorized. For example, this type of planning utilizes “honey pots” [30, 31] and threat modeling [32] for discovering potential adversarial activities and directing the adversary to system responses that provide misleading but plausible information. However, even areas populated by authorized users in well-controlled applications may be exploited to facilitate systemic denial and deception activities. An example would be a planning system that provides adversaries with the “wrong” plan, but one that would be compatible with other activities of which an adversary might be aware.

This section will deal with using honey pots for discovering potential adversarial activities and directing the adversary to system responses that provide misleading but plausible information. The example will be a planning system that provides adversaries with the “wrong” plan, but one that would be compatible with other activities that an adversary might be aware of.

Architectures designed for denial and deception activities may also include strategic use of both covert and overt channels in both deception and deception avoidance activities.

K. Shared Command Information Sharing

Shared Command Information Sharing	
Problem Statement	There have been an increasing number of events that involve multiple allies whose decision-makers have not traditionally been involved in matters of national importance. The increased policy focus on public-private partnerships is likely to further proliferate such scenarios. While current communications systems designs include services and support for security, such as access control, these initiatives have historically concentrated only on providing security within domains, not across them.
Background	In a world of ever increasing networked and service-oriented environments, separation of domains for security reasons has become impractical and runs counter to the need for information, service, and infrastructure sharing. Furthermore, technology adoption by current cross-domain sharing technologies is prohibitively slow due to certification and accreditation procedures.
Solution Criteria	New approaches to security engineering across domains are needed that can dynamically balance the need to protect information with the need to share.
Next Steps	In this module, command and control threat scenarios would be developed to provoke design choices from an adversarial viewpoint.
Thread	Frameworks

Dependencies	none
Timeframe	Mid term

On avenue of research relevant to this problem domain is that of cross domain information sharing, where domain refers to a system under the control of a single hierarchical command structure. If planned in advance, communications between domains may proceed with some level of trusted communication path during an actual emergency. This research would extend such advance-planning problem domains to include the ability to turn any set of command and control environments into a trusted communication environment by minimizing the extent to which they need to share information to accomplish mutual planning objectives criteria to a cross-domain command and control scenario.

Secure communications architecture is becoming increasingly important to, and entrenched in, military and intelligence operations, including initiatives such as net-centric enterprise services. Simultaneously, cross-domain technologies and solutions have begun emerging to handle the growing requirement to service the need to share information critical to military operations, disaster response, national intelligence, and other situations, as well as to balance the need to share with the traditional need to protect sensitive or classified information within and across domains.

There are a number of ongoing research efforts whose results may bear fruit for strategic cross domain information sharing, for example privacy-preserving communications protocols [33], collaborative wireless mesh networking, enabling discovery of service information, and managing identities and entitlements in the context of cross domain environments. Cross domain information sharing research is required to allow increased amounts of useful sharing without introducing vulnerabilities to confidential information. It includes identity mapping and entitlement models based on information usefulness to missions of interdependent organizations. Data-centric protection models are expected to be analyzed and tradeoffs between them quantified.

L. Physical Security Frameworks

Physical Security Frameworks	
Problem Statement	The impacts of managing requirement changes for physical security systems (PSS) requires end users and PSS providers to develop systems which can respond to known security threats as well unexpected emergent security threats.

Background	PSS is achieved by implementing comprehensive protection of high valued assets through the integrated and synchronized application of defensive countermeasures to mitigate risk. These measures include detection (sensors, network, communication), delay (physical barriers, network barriers), and response (DoD personnel) elements which are integrated into a *System of Systems* to provide the PSS service. It is also policy to extend the operational effectiveness of a PSS through application of appropriate risk management strategies, employ the most effective protection measures, to include system assurance and document the measures in a PSS Program Plan. The increased use of IT-based *Systems of Systems* to meet emergent threats has led to increasingly complex PSS which are more costly to implement, operate and maintain. The consequence is that high valued assets are not sufficiently physically secured in a timely manner, and not efficiently supported with a PSS lifecycle.
Solution Criteria	The solution would be a holistic but flexible approach to physical security requirements analysis that would take maximum advantage of emerging technology while providing state-of-the-art protective and detective physical security controls based on lifecycle and environmental characteristics of a system of interest.
Next Steps	An experienced PSS system provider in conjunction with an acquisition-experienced system security team will observe change control cycles within an existing DoD or similarly complex enterprise, recommend appropriate security engineering impact analysis on design and implementation of PSS solutions, and conduct an analysis of the impact that these recommendations will have on trade-space factors and increased PSS capability.
Thread	Frameworks
Dependencies	none
Timeframe	Mid term

Security metrics (Reference Section: 4.3)

M. Architecture Metrics

Architecture Metrics	
Problem Statement	There is no accurate, reliable, comprehensive way to measure information systems and infrastructure security. This study would recognize that security is a property of a system, a state that changes as the system evolves. System security metrics should be based upon the security solution design patterns and the specific risks that they are intended to reduced.

Background	The current approach to security metrics explicitly patterns criteria on security standards, the weakness of which have been discussed at the beginning of this section [34, 35].
Solution Criteria	A system security architectural formulation based on reusing security solution design patterns as the potential basis for a continuously expanding set of standard system security architectures for application by the system engineering community.
Next Steps	Identification of target architecture on which to base the approach. This may be combined with successful current approaches to measuring security configuration.
Thread	Metrics
Dependencies	Progress on Standards and Frameworks
Timeframe	Mid Term

In this module, methods and techniques for designing and assessing security architectures will be developed. The more generically the security requirement can be stated, the more options the systems engineer has for conducting trade-space analysis, and a larger number of architectural patterns may model the requirement.

Assuming that the design of the system followed the guidelines recommended in an architectural security pattern such as those discussed in the previous section on frameworks, corresponding metrics would be devised that would provide a clear indication of the strength of the security built into the design. For example, in the data continuity framework described above, it should be possible to relate the security provided by this design pattern to the false alarm rates caused by, discontinuities from other than malicious sources such as: a) possible tracking errors, b) sensor inaccuracies, c) data continuity quantization levels, d) data update rates, e) frequency for data continuity checking and f) the type and number of information delivery alternatives available to the end user/operator. The design pattern for continuity checking could include metrics that relate to each of these individual factors, as well as metrics that relate to the group as a whole. The integrated metrics could include ordinal as well as cardinal number-based metrics, with the security assessor selecting the most appropriate for the system and risks under evaluation.

This work would be done with the recognition that all systems of interest are open, therefore, systemic security changes over time. It would seek to establish qualities of systemic security metrics that:

- are outcome based
- provide an end-to-end assessment of systemic security
- appropriately bound vulnerabilities, and identify boundary movement
- cover recognizable hazards
- adapt with changing environments

Included in candidates for security architecture metrics would be:

- Ability to recovery from fault states
- Technology development regarding solutions that are closely related to specific aspects of a system.
- Supply chain trust considerations
- Analysis techniques for trade-off assessments

- Approaches for accounting for system-of-system considerations.
- Utility of red teams in support of evaluation and design activities related to security solutions.
- Operational impact of vulnerabilities
- Compartmentalization to minimize threat surface data or critical processing

Overall, the work is intended to promote more resilient security architecture by providing a means to measure changes in security posture based on new exploits, new capabilities brought on by software upgrades and new connections realized by the rapid reconfiguration of systems within a system of systems.

N. Risk Metrics

Security Risk Metrics	
Problem Statement	Incentive and motivation structures for owners and operators of critical infrastructure and privacy-obligated data. The characteristic that gives any system its potency, those parts of a system enhance the effectiveness of one another, also makes them susceptible to catastrophic failure if one of their central parts can be corrupted. Yet there are some aspects of the system-of-systems that ought to alleviate, if not refute, these concerns.
Background	Risk metrics in security has followed business risk analysis [36]. This approach instead focuses on risk to mission. Specifically, dependability models based on the divide between intrinsic and agile security should allow system integrity to be measured relative to security threats. The work of Khashanah in risk metrics will be complemented by the systems security risk framework of Ulibarri to create a quantitative approach to systemic security risk.[3]
Solution Criteria	The goal of this research would be accurate assessments of system strength relative to attack vectors. This would be a completely new way to measure security risk.
Next Steps	Development of a set of informational sensors necessary to span the system dynamics a sample of large-scale systems-of-systems. This would be followed by tomographs and tomography, risk set definitions, and diagnostic models.
Thread	Metrics
Dependencies	None
Timeframe	Mid term

In material science, “fracture critical” refers to the one beam that must bear weight in order for the structure to stand. If you know your system integrity, then the complement of that is your residual risk.

[3] Khashanah presented at the SERC Security Workshop (see Appendix C), Ulibarri attended the workshop and described her complementary approach during discussion periods.

Similarly, the goal of this research would be accurate assessments of system strength relative to attack vectors. Such research would entail:

- Development of a set of informational sensors necessary to span the system dynamics
- Development of tomographs and tomography
- Definition of endogenous and exogenous systemic risk threat sets
- A theory of security characteristics to describe the state of variables contributing to each of the characteristics
- Derivation of multi-scale dynamics
- Use of informational sensors and the imaging from tomography as input into the models
- Building a robust diagnostic system that identifies a stable evolution range
- Construction of an early warning system for potential systemic security issues

O. Security versus Convenience

Security Versus Convenience	
Problem Statement	When engineers are asked to design bolt-on security solutions, the results often decreases system usability. The question asked is not, "which feature should be preserved?" but rather, "is the security worth the cost and inconvenience?"
Background	It is the job of a systems engineer to specify what features may be preserved at the expense of others, to present hard choices with respect to system functionality and capability.
Solution Criteria	A systemic definition of convenience would not focus on human-computer interaction issues, but would instead strive for non-interference with system functionality. The trade-space with security would be the extent to which implementation of a suggested security mechanism would decrease system functionality. Ideally, it will motivate the introduction of additional security solutions at the systems function level.
Next Steps	This module is meant to address the trade-space issues with respect to convenience head-on. It should quantify how security affects human-computer interaction in ways that make sense at the trade-space level, and not as a post-design consideration.
Thread	Metrics
Dependencies	none
Timeframe	Mid term

The unquestioned security requirement based on standards approach seems finally to be tipping economies of scale in the wrong direction, providing ability for systems engineers to take a fresh look at security requirements. By determining security's impact on functionality at both the system and enterprise level, the security versus convenience trade-space becomes visible. Security features should

be examined in close coupling with the system of interest in order to determine if security goals are actually met, and if so, at what cost.

For example, were the authenticity feature above to be implemented on a system that did not have a prior login requirement, the system availability would be impacted by the number of seconds multiplied by the number of logins in any given time period. This decrease in system functionality increases as users forget passwords and there are delays in provisioning new users.

The key to convenient security is to decrease all unauthorized functionality while leaving the functionality that allows the system to achieve its mission. For example, a system that is designed to be a web information server does not have its functionality reduced by network placement behind a firewall that allows web services. There may be costs in administration and support for the network security measures, but these are not related to the system of interest, as it is narrowly defined.

Where the entire enterprise is viewed as the system of interest, direction of technical personnel to a security task may or may not reduce system functionality. In an environment that contains network-borne cybersecurity threats, an argument may be made that the firewall increases overall system functionality by reserving resources for internal processing that otherwise would be exploited by joy-riders as well as random viruses and worms. Yet, if it is determined that the web services are not susceptible to stateless overt channel attacks, then a router may suffice to thwart these threats, and the opportunity costs of protection may be reduced. The trade-off may still be characterized as convenience, as convenience in administration is increased by decreasing the number of administered devices.

This approach to security analysis is at once old and new. In the early days of eCommerce, security requirements were often stated as in our example (i.e. *"user identification and authentication shall not take more than 30 seconds"*). This is how the plethora of security products on the commercial market got their start. Unfortunately, due to common problems in cybersecurity combined with the current ubiquity of cyberspace, the existence of these security products have since skewed the buy versus build cost-benefit analysis for security features very far in the direction of buy. A growing appreciation for cybersecurity standards has introduced requirements for bolt-on security devices in a variety of situations that would not otherwise have motivated an enterprise to include them in their networks.

One example is intrusion detection. Off-the-shelf intrusion detection systems monitor network traffic for a large set of publicly known attack patterns. These patterns are also available to anti-virus vendors, and most of them have corresponding operating system patches. So in an enterprise where patches and virus patterns are up-to-date, the detection of an attempted intrusion provides very little value-add for the amount of technology resources it takes to deploy such a system network-wide. This decreases the overall functionality of the IT organization. Contrast this with the ability of the enterprise to invest the same amount of money in application-level fraud detection. With such an investment, they would likely detect attacks that had successfully evaded their preventive controls. Assuming that both the intrusion detection and fraud detection take the same amount of deployment resources, and that the cpu or

network cycles may also break even, the trade space with convenience defined as core functionality would find no functional decrease in either. However, application fraud detection actually increases system functionality at both the system of interest level and the enterprise level, because it presumably would uncover evidence of actual fraud and so allows the enterprise to prevent further damage and possibly recover stolen assets. In contrast, infrastructure intrusion detection merely identifies suspicious infrastructure-level behavior, rather than actual evidence of harm to the enterprise. It prompts incident investigation, which is more work, and may never be correlated with actual harm.

A research program based on the premise that security impacts convenience would examine enterprise level security controls in the context of system functionality. Given a sample enterprise or set of enterprise cohorts, the security versus convenience trade-space could be examined both the application and enterprise functionality level both with and without current or planned security features. Most enterprises of a significant size will have a ready starting point for this type of investigation because they have security-related responsibilities, process and technology described in a documented security program. Interviews could be conducted to validate that significant elements of the program actually deliver key security features as documented. The trade-space could be quantified in terms of both application and enterprise opportunity cost without sacrificing the utility of threat avoidance. The economics of various security feature implementation strategies could be compared across an application inventory and also at enterprise levels.

P. Security Trade Spaces in Emerging Technologies

Security Trade Spaces in Emerging Technologies	
Problem Statement	Roadmap elements should include focus on future threats and opportunities.
Background	It is expected that advances in technology will provide an increasing number of security engineering challenges. Multicore chips will involve more concurrency vulnerabilities, but also opportunities to use some of the CPU cores for security monitoring, analysis, deception, deterrence, etc. Cloud services will be harder to reverse engineer. Autonomic systems will have limited commonsense reasoning and spoof-resistance capabilities, implying the need for human monitoring and mixed-initiative approaches to their security. Systems of systems will need to deal with larger and larger numbers of independently-evolving co-dependent external systems, implying the need for incremental vs. start-from-scratch security analysis capabilities.
Solution Criteria	Frameworks in section 3 are concerned with existing problems, but metrics should be able to spot evolving trends in ways that allow shared concepts about future technology capabilities that may be necessary to make transparent decisions about tradeoffs between cost and functionality of various security features or implementations of security features.
Next Steps	This module has a dependency on both systems engineering security

	standards and security architecture. This module is expected to be informed by output from the standards, architecture and risk metrics modules, and would not be attempted without successful results in those areas.
Thread	Metrics
Dependencies	Standards and Architecture Metrics
Timeframe	Long term

In addition to the growing threats of attack is the corresponding increase of consequences that can result from cyber security incidents due to the increasing integration of network technology into systems that previously did not rely upon it. For example the convenience and savings benefits of radio frequency identification cards (RFID) has made their use widespread, from credit cards to security badges, but inexpensive methods exist to steal and clone the private information stored on them; and the corresponding risk to many large organizations is serious. Convenience and cost savings provided by networked systems are two drivers of increased cyber security risk, and will likely continue to increase the need for enhanced cyber security capabilities.

Q. Trust Assessment Models

Trust Assessment **Models**	
Problem Statement	With increasing frequency, the government and its commercial supplier base rely on foreign companies to produce the most advanced technology solutions. Once dominated by domestic manufacturing, today's technology manufacturing is largely conducted outside the United States. Product development in both hardware and software is thus subject to supply chain threats in both construction and operation.
Background	Trust metrics candidates under consideration in today's literature include, but are not limited to, size of community, symmetry, transparency, degree of control, consistency of presentation, integrity, offsets, value of reward, components, and porosity [37].
Solution Criteria	Techniques made available to systems engineering that will provide the most current and accurate attribution of trust. These should be capable of being weighed in the context of an evidence framework, while avoiding pitfalls due to composability and transitivity
Next Steps	A survey of MPTs available to systems engineering that address authenticity and source attribution as well as investigation into hardware and software trust avoidance mechanisms. Standard efficacy nomenclature for trust metrics.
Thread	Metrics
Dependencies	none
Timeframe	Short

This module combines comments during the conference on the topic of trust. In building systems, systems engineers often specify components without specifying components security properties. Functional components with insecure heritage often present security risks. However, heritage is not always possible to specify, sometimes for logistics reason and also because sometimes systems components are already in place and cannot be expected to be specifiable at system design (e.g. a user desktop). Where trustworthy components are necessary to overall mission assurance, the trust level must be assured by inspection. This research proposes methods for systems engineers to specify trust inspection models. These would apply not only to systems components but to systems interfaces and counterparty systems.

This module depends on the risk metrics in that systemic risk components and the residual risk of counterpart reliance would have to be taken into account when assessing trust. It depends on frameworks modules to report metrics as they encounter trust criterion useful for components evaluation in their respective environments. The coordination module will be particularly critical in ensuring that supply chain risks are adequately covered in the earlier modules in order to provide meaningful architecture metrics and examples of potential for security integrity in the presence of untrusted components.

What is needed is an approach to system engineering in which imperatives for trustworthiness based on various criteria, are a deeply and thoroughly embedded into all life cycle phases of the system – i.e., specification, modeling/simulation and architecture, design and implementation, test and operation – as the need for functional correctness. What is needed is an adaptable approach to modeling, measuring, and assuring trustworthiness regardless of the system engineering methodology being used. The result will be a system engineering methodology that ensures and enables the specification of trustworthiness-attaining and assuring (through measurement-based verification) requirements, the verification that system architectures/models exhibit all specified trustworthiness properties at their required level of assurance, and the verification that the system continues to exhibit those properties at their required level of assurance through each more detailed iteration of design and implementation.

These trust models and metrics will ultimately allow systems to operate in the context of a current and accurate attribution of trust. The security functionality that implements trust measurement will be usable both in the engineering of the system (at all levels of development), as well as in its testing and independent assessment, and during its operation. For example, the system in operation would be able to use these techniques to verify its own trustworthiness according to standard criteria were used to define that trustworthiness. For example, the integrity of configuration and interfaces, strength of self-protection against state changes resulting from anomalies or other events would be expected to be applied to the system of interest or to its communication or interaction with other systems that employ the same or similar techniques. As noted, the techniques would be applicable to a wide variety of system models and architectures, while avoiding pitfalls due to composability and transitivity.

In this new approach, specification, architecture, and design of the system will incorporate not only system features but the ways and degrees to which those features must be able to be trusted. Moreover, the approach includes a method for establishing metrics for measuring trustworthiness (as the basis for assurance) throughout the system life cycle.

It should facilitate fast development of trusted systems by providing parameters with which to measure the benefits with respect to trust of security features. The solution must be flexible and adaptable to a variety of systems architectures.

The approach will also provide a basis for streamlining the independent certification of system trustworthiness by providing a basis for rethinking the whole current approach to system C&A, by allowing continuous verification and reverification of trustworthiness of each level of system engineering artifact (specification, architecture, model, design, implementation, etc.) as it is being produced, rather than delaying such certification until after the conclusion of a life cycle phase.

The solution will be flexible enough to work effectively across the wide variety of current system models and architectures, and engineering methodologies, and adaptable to continue working effectively with emerging models, architectures, and methodologies, thereby ensuring that trustworthiness is engineered into systems regardless of how those systems are conceived and built, and ensuring that systems as widely ranging as information systems, software-enabled/embedded systems, and physical systems are inherently, measurably, and assurably trustworthy.

There are two ways to view state of the practice in this field. Identity and authentication trust models usually rely on customized identity and authentication techniques that take advantage of trusted platform module (TPM) features of COTS chips [38]. Systems operational trust models usually rely on change control, data integrity checking, system responses to vulnerability assessment, and/or security log analysis [39].

The state of the art in trusted systems development is hardware chips that segregate memory in response to commands issued by customized software processes [40]. Common applications for these techniques are identification, authentication, and encrypted communication. This type of model could be extended into a trust library that made use of the hardware-embedded keys to verify trust properties such as the integrity of system configuration and software binaries. A combination of verified measures would form a trust metric. Trust metrics candidates under consideration in today's literature include, but are not limited to, size of community, symmetry, transparency, degree of control, consistency of presentation, offsets from expectation, and component integrity [8, 37, 41]. The type of trust model proposed has two components: trust evidence and trust computation [42]. Both sets of features require research to ensure adaptability, but a flexible model using state of the practice techniques for trust evidence and state of the art techniques for computation should be possible to develop fast. This approach has been applied to demonstrate that metrics for flexibility in space systems engineering can be used to establish design criteria that will allow maximum utility during the lifecycle of a system [43]. Similarly, a base set of metrics could be established that enable some trust, and additional metrics may be added as they are developed.

This module should reflect on work at DARPA and commercial industry regarding trusted components. It would include new and innovative approaches to signature generation and verification, invisible watermarking, software flaw detection, techniques for secure and reliable computational outsourcing, predictive blacklisting, and techniques for supply chain collaboration among trusted partners.

Security workforce (Reference Section: 4.4)

R. Workforce Education

Workforce Education	
Problem Statement	Although some literature on security architecture and engineering exists, it does not actually include instruction on security architecture and engineering at the detail necessary to design and architect secure systems.
Background	There are few textbooks devote to the subject of Security Engineering. Perhaps the most popular is over 1000 pages yet devotes only 6 of those pages to design methods and covers the subject from the point of view of management rather than engineering[44]. Another treats the topic of enterprise security architecture at too high a level to be useful to a systems engineer[45].
Solution Criteria	This module should result in a SSE curriculum that may be delivered to an engaged workforce.
Next Steps	This module should compile available knowledge on the topic of MPTs in SSE. It is also expected to identify and explore issues related to appropriate education delivery methods. These include range of knowledge required for a given task or domain, and length of viability of certain types of information.
Thread	Workforce
Dependencies	Standards and Metrics
Timeframe	Mid term

Next steps in pursuing improvements in workforce training were identified as requirements, architecture, technologies, concept of operations, and skills sets.

Requirements: Any good system design begins with requirements definition. The same applies to making a system secure from the start. In addition to defining functional requirements, the requirements definition phase must involve identifying the requirement for the three basic security properties of confidentiality, integrity and availability. Questions about the requirements for each of these properties should be cast along the lines of the 5 W's of who, what, when, where, and why, and it is particularly important that they be answered also in the negative, who should not… The requirement definition phase includes individuals and stakeholders with non-technical backgrounds. We will need resources to inform not only the technical staff involved in system design, but also military staff who might rotate into a program office from an operational command, or to inform legislative staff that might be responsible for understanding requirements that might be spelled out in legislation authorizing specific programs.

Architecture: Securing a system requires attention in the architecture phase. In a good design, the developers create a high level specification of information and control flows. It is here that the security requirements get mapped to something concrete. The high level components of the system and the flows are described in the system design and the constraints on the flow of data need to be taken into account. Here, one should define the allowed flows across and between components, and any flows to / from components external to the system. The requirements should be applied in validating the flows, and an output of this activity should involve a list of components and acceptable flows in machine readable form. This is one area where new tools, or extensions to existing tools, can be created that will force designers to take these steps. Training will be needed and over time, such a process needs to become a basic part of the skills taught for system design to all computer science students.

Technologies: Security practitioners will continue to play a role in securing systems, and the methods describe above will help them to better integrate the traditional security technologies into new systems. The flow constraints specified by the system engineers, architects, and developers must be converted into policies that are enforced within access control and other mechanisms.

ConOps: Such a basic understanding of the security of a system must also be carried through into the operational phase of a system. The basic educational training for those that will operate such systems needs to include aspects of security, and the system specific training must instill an understanding of the security philosophy of the system. This "philosophy" needs to include issues such as separation of roles as an approach to mitigate insider threat. These approaches need to be understood in addition to the specific checklists that are more commonly used in such settings.

Skill sets: The design of the system itself must also include a specification of the skills that will be required of the operators of the system, and must include training for the operators on how to maintain the security of the system and respond to detected security events.

This module is dependent for curriculum on the completion or near completion of modules in security standards and architecture. It is dependent for execution on the career path module as that module is expected to validate or further identify the full realm of job functions that require SSE training.

S. Security Requirements Process

Security Requirements Process	
Problem Statement	Systems security requirements are difficult to establish and justify.

Background	Current system security engineering processes focus on risk-based impact analysis and systems development lifecycle issues to catch potential vulnerabilities in systems under development. Security specialists are assigned to projects to ensure that known threats and vulnerabilities are covered by appropriate security controls. This creates a resource issue as assigning a security expert to every systems engineering project has often been proposed as a solution to this problem.
Solution Criteria	Solutions must address the ability of a systems engineer to recognize when faults in systems security requirements gathering efforts occur in time to raise a flag that assistance may be required.
Next Steps	The research would be expected to identify and codify known, and apparently successful, SSE process into a language that would allow them to be analyzed for faults and single points of failure.
Thread	Definition, Workforce, MPTs
Dependencies	none
Timeframe	Short and medium

SSE has often been called an art rather than a science because of the specialized knowledge required for its practice. Oren has described an effective and efficient process that is broadly understood within her agency. It requires that a security systems engineer with specialized security skills be an integral part of the systems engineering process. Routinely applied, it achieves the security required by today's systems. The process addresses security considerations into systems at all stages of development.

Osterweil and Clark have demonstrated the use of process definitions and analysis to create a framework within which system security issues can be identified, and solution approaches evaluated, to support continuous security improvement. Many of a system's security vulnerabilities can be identified by studying the processes that will use the system. These processes are typically collaborations among a variety of types of agents, some of whom may pose security risks. Some of these risks can be defended against by appropriately designed and implemented processes. Representations of these processes, sometimes coupled with representations of the behaviors of both attacking agents and defending agents, can be studied through adaptations of existing software analysis approaches to determine whether defined process features and defense behaviors are suitable for thwarting the modeled attack behaviors.

This module combines the systems engineering process documented by Oren with the process analysis framework described by Osterweil and Clark. [4] The goal would be to validate the Oren model for systems

[4] See Appendix C: *SERC Security Engineering Workshop Agenda*

engineering security, or to identify gaps or weaknesses which could then be a subject of recommendations for next steps.

The combination of these two research approaches would use Oren's systems engineering process itself as an object for the process analysis framework described by Osterweil and Clark. As a systems engineer works with stakeholder on a systems engineering project, it is subject to the same threats and faults as any system in operation. A direct application of the Osterweil and Clark process analysis framework to the Oren system security engineering process would shed light on the feasibility of relying on security-specific process frameworks in the context of the overall systems engineering endeavor. The results of this study would inform the Career, Metrics, and SE Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE) modules.

T. SE Career Path

SE Career Path	
Problem Statement	SSE is not part of any well-define career path. Talented individuals are not actively recruited into the field.
Background	This module combines comments during the SERC Security workshop on the absence of well-defined career paths in systems engineering in general, and security engineer as even more or a niche.
Solution Criteria	The research is expected to explores incentive for including security in systems engineering job functions as well as stakeholder and associated workforce multiple job functions such as management, development, and operations.
Next Steps	It is also expected that that module will highlight process roles and responsibilities performed by various actors in SE Process. These results would be used to create a survey to be circulated among systems engineering groups that have successful security results. These best practices would be vetted via focus groups and refined to propose a systems engineering career path that rewards and enriches security-enabling activities and decisions.
Thread	Workforce
Dependencies	SE Process
Timeframe	Short term

It will be informed by the Systems Engineering Process Fault Identification module as that module will make the interaction between the security systems engineer and an example systems engineering process explicit.

Security MPTs (Reference Section: 4.5)

W. Exploring Nearby Disciplines

Exploring Nearby Disciplines

Problem Statement	There are many MPTs that are well established in disciplines that are related to security or have similar goals or objectives. Successful MPTs in these areas should be examined for possible application to systems security.
Background	The Joint Software Systems Safety Handbook[5] was developed by a group of software safety experts from DoD and other Government agencies in cooperation with industry. The handbook will be available for use by the DoD and wider community, also on industry and academic sites. Recognition that the MPTs in this tool would be applicable to system security motivated the possibility of further exploration of security-related disciplines.
Solution Criteria	New ways of approaching security architecture and metrics issues, as well as new frameworks within which to identify security patterns.
Next Steps	Without initially focusing on any one tool, a research team would identify and analyze related disciplines like Safety, Reliability, and Surety in search of MPTs that have relevance in system security engineering. These tools would then be prioritized for further study as to their utility in the security space.
Thread	Workforce
Dependencies	none
Timeframe	Short term

Agency departments and industry are expected to collaborate on updating and agreeing on latest best practices. Processes and best practices for modern software implementation (e.g. networks, open architecture, system of systems) are being developed. Flow charts with entrance and exit criteria for software safety activities are being defined, these would be monitored, and successful applications incorporated in metrics and education modules.

Some of the costs associated with achieving higher levels of security are the losses in cost-effectiveness often caused by over-optimizing on security at the expense of other objectives. Examples include decrements in reliability and safety via single-points-of-failure such as single-agent key distribution systems or non-replicated data; decrements in performance via high-overhead security defenses; decrements in usability via user-interface overhead or password-protected assets that can't be used by other soldiers when on gets incapacitated; and decrements in evolvability via overly slow recertification procedures. On the other hand, some related discipline investments can increase cost-effectiveness, such as for integrity and monitoring capabilities.

X. BKCASE Security Section

BKCASE Security Section

[5] See Appendix C: SERC Security Engineering Workshop Agenda, presentation by Arch McKinley.

Problem Statement	Systems designs and operating environments are increasingly required to comply with a wide variety of complex security standards documents. Systems engineers currently independently reconcile such standards to system requirements on a project by project basis, resulting in countless hours of rework without reuse. The complexity renders the current security requirements process not only inefficient but error-prone.
Background	BKCASE (pronounced "Bookcase") is the acronym for the Body of Knowledge and Curriculum to Advance Systems Engineering. The project scope is to define a Systems Engineering Body of Knowledge (SE BoK) and use the SE BoK to develop an Advanced Graduate Reference Curriculum for Systems Engineering (GRCSE, pronounced "Gracie") [46].
Solution Criteria	A unified model for security standards that is comprehensible to systems engineers and easily applied to a wide variety of systems.
Next Steps	The research would be expected to catalogue and model 15-20 influential security standards into a common systems engineering modeling tool. It would produce a guide for systems engineers to use to ensure that they are compliant with one or more specific standards.
Thread	Definition
Dependencies	Security Standards Reconciliation Security Requirements Process
Timeframe	Short-term

As the overall goal of the roadmap is to produce MPTs that will be widely applicable for systems engineers primary guidebook for the systems engineering community going forward has been established by both public and private sector stakeholders to be the SE Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE) [46]. Research modules that identify useful elements of currently available security standards and best practices will be expected to contribute to systems engineering security education, and hence are expected to culminate in contributions to the BKCASE knowledge repository. Like the ISO standard on system assurance, this document was created by engineers and for engineers, and is expected to be a familiar and trusted reference for the SE community. The contribution of sound security guidance to BKCASE would be the most significant deliverable for this roadmap. However, this achievement will only be possible once the plethora of other security standards of value are consolidated into a comprehensible body of knowledge, and supplemented with out-come based architectural strategies that will improve the ability of a systems engineer to achieve measurable security results. Hence, the Security BKCASE research module is depicted as dependent on modules relating to standards consolidation.

Security advanced topics (Reference Section: 4.6)

Y. Agile Architecture

Agile Architectures	
Problem Statement	Today's large and complex systems already exhibit massive information exchange, unpredictable coupling both internally and as systems of systems components. Emergent functionality has the appearance of swarm intelligence and living loops in fast evolution. To preserve critical program information and mission assurance, the security of these systems must also exhibit quick reaction capability and rapid reaction and capability acquisition.
Background	Systems thinking words to describe features to anticipate unknowable and therefore unexpected interaction with other systems are agility and adaptability. These describe a system that contains inherent non-equilibrium and processes information from internal and external sources as feedback to enable change and growth. This is best understood in contrast to a system that simply processes information in predefined ways to achieve specific predetermined outcomes. Such systems are frequently described as self-organizing, as they possess ability to reorganize their internal state, interfaces, or other functional components in response to new information [47].
Solution Criteria	A systems thinking approach suggests that a system's level of agility and adaptability may contribute to its response-ability to the changing operational environment and threats. Solutions in this space would identify system features, associated architecture and metrics that would provide system agility.
Next Steps	This module will proceed in conjunction with the INCOSE security working group proposals to identify and instantiate agile architecture designs.
Thread	Advanced
Dependencies	None
Timeframe	Long term

This module recognizes the potential of the INCOSE Security Working group's pattern-based approach to security [48].[6] In these patterns, threats are recognized by independent systems components, each of whom responds in a way that provides the system with threat mitigation response. Individual agent functions include peer-monitoring and self-organization. This research would investigate executable architectures at both the agent and pattern based level.

[6] See Appendix C: SERC Security Engineering Workshop Agenda, presentation by Rick Dove.

The definition of pattern according to INCOSE, and in this section, differs slightly from that in section 4.2 on Frameworks. The Frameworks section does not specifically require all patterns to exhibit agile characteristics, and this requirement introduces a measure of complexity in implementation that may perhaps introduce an unnecessary architectural constraint. However, this module has the potential to inform the architectural metrics, command as well as the framework security modules. It will focus on the utility to systems engineering of at least six shared agile-system characteristics:

- Self-organizing – with humans embedded in the loop, or with systemic mechanisms.
- Adapting to unpredictable situations – with reconfigurable, readily employed resources.
- Evolving in concert with an ever changing environment – driven by vigilant awareness.
- Resilient in reactive response – able to continue, perhaps with reduced functionality, while recovering.
- Innovative with proactive initiative – acting preemptively, perhaps unpredictably, to gain advantage.
- Harmonious operations – aiding rather than degrading attack-system functional productivity.

These have been selected because they are currently properties of intelligent attacking systems. It is expected that security features may be developed that mirror the agile attack community's characteristics. A wide variety of research in threat detection is available as starting point for these studies.

Where a system exhibits these characteristics, it will have achieved a measure of security that supports mission assurance. These will be defined by pattern and measured according to the evidence of standard pattern elements. INCOSE patterns are described contextually using the following characteristics.

Name: Descriptive name for the pattern.
Context: Situation to which the pattern applies.
Problem: Description of the problem.
Forces: Tradeoffs, value contradictions, key dynamics of tension and balance, constraints.
Solution: Description of the solution.
Graphic: A depiction of response dynamics.
Agility: Evidence of characteristics that qualify the pattern as agile and adaptable.
Examples: Referenced cases where the pattern is employed.

Z. Executable Architecture

Executable Security Architectures	
Problem Statement	Attack vectors change with operating environment. Security measures should be chosen based on their ability to thwart attacks on the system of interest.
Background	Agility and cognitive computing models have common elements in that they both result in security as a property of a system in operation, rather than security as a feature that is bolted onto system functionality.

Solution Criteria	This study should analyze and document patterns of attack vectors and corresponding systems security architecture recommendations. It will study the efficacy of active defense models in achieving goals for mission assurance. It will integrate findings in several other research areas as they become available in order to produce a comprehensive approach to mission assurance frameworks that include attack damage assessment, technology protection, supply chain risk, and resilient software processes.
Next Steps	Research in cognitive computing will be examined to identify possible reuse of agents designed to collaborate using economic exchange, cooperation, and mutual protection models. These models incorporate sensory input, perception, reflection, analysis, and also incorporate learning mechanisms to inform future action.
Thread	Advanced
Dependencies	none
Timeframe	Long term

Cognitive computing typically concentrates on individual agent-based and includes ability to assess, analyze, and respond to the environment [49]. Agile and cognitive capabilities are typically modeled as multi-agent systems based on algorithms that make use of trade spaces such as game theory. Agents are represented both individually and in aggregation, as cognitively aware combinations of people, processes, and technology capabilities. Capabilities are formally modeled. This research module will examine the feasibility of building such models in an environment that combines hardware and software components in support of functional protection or mutual protections models in system-of-system scenarios.

A common systemic security analogy for cognition lies within the context of a command and control structure's observe, orient, decide, and act process (an OODA loop). One of the most influential adopters of this approach was the military strategist John Boyd, who stated, "without OODA loops...we will find it impossible to comprehend, shape, adapt to, and in turn be shaped by an unfolding, evolving reality that is uncertain, ever-changing, unpredictable" [9].

An OODA loop assumes the existence of a decision-making entity as a critical system component. Decision making entities may be man or machine. Their decisions are likely to be enabled by some type of decision-support system that converts data about the system and its environment into descriptive, metrics-based information designed to identify changes. The relative timing of threat, attack, and response can be explored as interwoven OODA loops (as adversaries try to get inside a responder's OODA loop, and vice versa.)

In a response-able system, knowledge sharing and visibility across the enterprise will be sufficient for stakeholders to make effective strategic- and implementation-level decisions about system design, development, and configuration decisions. Stakeholder decisions will set up the conditions needed for

success and the system architecture will be designed to implement those decisions and respond effectively to directed changes. The pattern is similar at higher levels of agility: a self-organizing system will adapt and respond to strategic-level stakeholder decisions without the need for specific, prescriptive implementation-level decisions; a complex adaptive system will learn and adapt to the results of ongoing operations and from internal and external information resources to change as needed over time to meet the system's mission requirements and maintain the identity and coherence of the system.

The degree to which internal components of a system enable, or may be redesigned to improve, response-ability will determine its emergent systemic security. This research will attempt to identify architecture designs that rely on self-organizing capabilities or controls to improve the systems' overall response-ability using self-organizing behaviors of the system in its operational environment. For example, it would build on current research in ad-hoc networks, diversification and randomization, as well as automated intrusion response.

AA. Critical Functionality

Critical Functionality	
Problem Statement	Where systems have multiple stakeholders and sometimes conflicting requirements, a clear idea of core mission would be helpful in prioritizing security efforts. As the Apollo capsule fell to earth, it jettisoned unnecessary equipment in order to achieve the balance necessary for reentry. Similarly, a clear focus on mission assurance should be helpful in making security sacrifices while under threat from a determined adversary in order to maintain critical functionality.
Background	Though this focus on mission was not part of any other workshop component, it was obvious upon reviewing the workshop themes that focus on the *core elements* a mission should be a critical component of the quest for the core of the systems engineering security trade space.
Solution Criteria	This module would be expected to provide patterns and trade-space metrics that enhance trust and capability of mission critical functions and provide for continued operation of these functions in the face of damage to those of lower priority.
Next Steps	Sample problem spaces should be chosen examined for similarities that would allow for initial models of core mission capability versus secondary or supporting functions.
Thread	Advanced
Dependencies	none
Timeframe	ongoing

Security Research Coordination (Reference Section: 4.7)

BB. Coordination

The responsibilities of coordination will be to ensure communication among researchers and progress toward overall roadmap goals.

Communication	
Problem Statement	There are a variety of overlapping research tasks within this roadmap and diligence should be exerted to ensure that overlapping projects share research results in a timely manner, that the same research module is not repeated in different projects, and that publications resulting from research studies are made generally available to appropriate SE community stakeholders.
Background	As the SE UARC is the official vehicle for systems engineering research in the Federal Government, and SSE is a topic of wide applicability, the program should include assurance that research results will appropriately benefit all government stakeholders.
Solution Criteria	There shall be transparency in research progress and vehicles for communication between members of the SERC security community as well as periodic infusion of ideas emerging in security research outside the SERC. Publication forums and information sharing activities will be essential to the success of this effort.
Next Steps	A SERC security roadmap committee should be formed and given the charter to socialize the SERC security research roadmap, coordinate research efforts within it, and maintain awareness of systems engineering research efforts with similar goals and objectives.
Thread	Coordination
Dependencies	none
Timeframe	ongoing

CC. Hypothesis Test

Test of Hypothesis	
Problem Statement	This roadmap is intended to increase the proficiency of systems engineers when it comes to security.
Background	As described in section 3, the solution criteria for the success of this roadmap is more proficiency in security among systems engineers, whether it be individually or in teams that produce more secure systems. The motivation for this effort is a need among US government sponsors for more secure systems.
Solution Criteria	A test of security awareness for systems engineers who are the main targets of this study.
Next Steps	In conjunction with the first study launched within this research

	roadmap, a combination of surveys and audit results should form the baseline for an opinion of security proficiency. These should be repeated periodically after a significant portion of the research modules are completed.
Thread	Coordination
Dependencies	none
Timeframe	ongoing

This module is expected to be a sanity check on the progression of the research roadmap itself.

Appendix B: Glossary

Architecture: A complete description of system design, including a logical model of functional decomposition allocated to physical resources (as Buede defines *allocated* architecture [50]).

Assurance: Grounds for justified confidence that a claim has been or will be achieved (From ISO/IEC DTR 15026-1).

Assurance case: Representation of a claim or claims, and the support for these claim (From ISO/IEC DTR 15026-1).

Commercial off-the-shelf (COTS): Technology components such as computers, communications equipment, integrate circuits, and application software that are available for purchase from one or multiple vendors.

Covert Channels: use TSEC definition

Domain: A segment of a network devoted to the information processing requirements of a given community.

Information Communications Technology (ICT): includes, but is not limited to, information technology (IT) as defined in title 40, U.S. Code (U.S.C.), section 11101. This term reflects the convergence of IT and communications. ICT includes all categories of ubiquitous technology used for the gathering, storing, transmitting, retrieving, or processing of information (e.g., microelectronics, printed circuit boards, computing systems, software, mobile telephony, satellite communications and networks).

Critical Program Information (CPI): ICT that is a critical component is defined as Critical Program Information (CPI) under DoD Instruction (DoDI) 5200.39, Critical Program Information (CPI) Protection Within the Department of Defense, July 16, 2008.

Cyber Security: Measures taken to protect a computer, networks, or information or computer system (as on the internet) and electronic information storage facilities belonging to, or operated by or for, the DoD or US Government, against unauthorized access, or attack, or attempts to access (DoDI 5205.ff: Defense Industrial Base Cyber Security/Information Assurance Activities).

Emergent Property: An attribute exhibited by a whole that is not attributable to its parts.

Framework: The concept of operations, mission, and environment under which a system operates.

Information Assurance: Measures that protect and defend information and information systems by ensuring their availability, integrity, authentication, confidentiality, and nonrepudiation (DoD 8500.01E: Information Assurance).-

Highly Optimized Tolerance (HOT): A property of systems whereby reaction to known threats is predictable enough to be itself a vulnerability.

Method: A collection of inter-related processes, practices and tools. A method is essentially a "recipe." It can be thought of as the application of inter-related processes, practices and tools to a class of problems with something in common.

MPT (Method Process Tool): A systems engineering technique that combines methods, processes, practices, procedures, guidelines, and tools to achieve system objectives. A *useful* MPT is defined as one that is:

- Relevant to the application environment: applicable to some subset of systems within the target environment.
- Repeatable: sufficiently well defined that implementation is possible in a different context.
- Likely to have significant impact: can materially improve systems engineering practice in the application environment.

A *viable* MPT is successfully implementable in the target organization given appropriate and reasonable tailoring.

Overt Channels: Methods of tunneling or otherwise hiding malicious network traffic in communication paths intended for other, more benign, protocols.

Practice : Activity that defines how to accomplish a task. (The terms “practice,” “technique,” and “procedure” are often used interchangeably in disciplines such as systems engineering). The tasks associated with a process are performed using practices.

Process: A logical sequence of tasks intended to achieve an objective. The objective achieved may be abstract (e.g. “negotiate among multiple stakeholders”) and/or a composite of multiple individual goals (e.g. “Deliver a fixed-date, variable-scope system”). The structure of a process enables levels of aggregation to allow analysis at multiple levels of abstraction in support of decision-making.

Security: Something that thwarts perpetrators who enact threats that exploit system vulnerabilities to cause damage that adversely impacts system value.

Security Feature: A system capability that contributes to its security.

Security Metric: Measurement that characterizes an attribute of the system of interest that is proposed to have both face and construct validity in the context of a hypothesis that the system is secure.

System: A model of an entity characterized in terms of hierarchical structure, emergent properties, and command and control.

System Assurance: The justified confidence that the system functions as intended and is free of exploitable vulnerabilities, either intentionally or unintentionally designed or inserted as part of the system at any time during the life cycle (NDIA Engineering for System Assurance Guidebook)

Systems Engineering: An interdisciplinary approach and means to enable the realization of successful systems. It focuses on defining customer needs and required functionality early in the development cycle, documenting requirements, then proceeding with design synthesis and system validation while considering the complete problem, including system operations, cost schedule, performance, training, support, test, disposal, and manufacture. Systems Engineering integrates all the disciplines and specialty groups into a team effort forming a structured development process that proceeds from concept to production to operation. Systems Engineering considers both the business and the technical needs of all customers with the goal of providing a quality product that meets the user needs (www.INCOSE.org)

System Security Engineering: An element of system engineering that applies scientific and engineering principles to identify security vulnerabilities and minimize or contain risks associated with these vulnerabilities (MIL-HDBK-1785: System Security Engineering Program Management Requirements)

Systems Thinking: An epistemology based upon systems concepts, characterized in terms of hierarchical structure, emergent properties, and command and control.

Tool: Something which automates or partially automates a specific practice or process, and thereby enhances task performance efficiency. The purpose of a tool is to facilitate the accomplishment of the process task. Some tools used to support systems engineering are model-aided.

Appendix C: SERC Security Research Workshop Agenda

Wednesday, March 31, 2010

Day 1

7:30 am - 8:30 am Registration – ***Stevens Institute of Technology office***, ***Ground Level***
7:30 am - 8:30 am Continental Breakfast – ***Conference Room Hemisphere B*** –**Concourse Level** – (go down escalator one level)
*Day 1 sessions are in the **Hemisphere B Conference Room, Concourse Level** unless otherwise noted*
SYSTEMIC PERSPECTIVE
8:30 am - 9:00 am Keynote – Kristen Baldwin, OUSD(AT&L), Office of The Director, Defense Research and Engineering (DDR&E)
9:00 am - 9:30 am Keynote – Dennis Barnabe, National Security Agency
9:30 am - 10:00 am Security Definition - Jennifer Bayuk, Stevens Institute of Technology
Break
10:15 am - 11:00 am Security Framework – Barry Horowitz, University of Virginia
11:00 am - 11:45 am Security Metrics - Drew Hamilton, Auburn University
11:45 am - 12:30 am Security Workforce - Clifford Neuman, University of Southern California
12:30 pm -1:15 pm *Lunch, Stevens Institute of Technology, Ground Level*
1:15 pm - 1:45 pm Illuminating Next Generation Patterns of Agile Systems Security – Rick Dove, Paradigm Shift International
1:45 pm - 2:45 pm *Panel*
A Process-Based Framework for Continuous Improvement of System Security - Leon J. Osterweil, Lori Clarke, University of Massachusetts
Metrics and Framework - Raghu Sangwan, Pennsylvania State University
Marine Security - Jose Ramirez-Marquez, Stevens Institute of Technology
2:45 pm - 3:30 pm *Discussion and Break*
DOMAIN PROBLEMS - FRAMEWORKS
3:30 pm - 4:00 pm Cross Domain Information Sharing, Michael Atighetchi, Joe Loyall, Partha Pal, Raytheon
4:00 pm - 4:30 pm Video and Audio Security Frameworks – Peter Beling, University of Virginia
4:30 pm - 5:00 pm Forecasting Systemic Risk for US Financial Markets – German G. Creamer, Khaldoun M. Khashanah, Stevens Institute of Technology
5:00 pm - 5:45 pm *Discussion*
5:45 pm - 6:00 pm Closing Remarks Day 1: Jonathan Goodnight, OUSD(AT&L)/DDRE/SE
6:00 pm - 7:00 pm Reception – ***Stevens Institute of Technology, Ground Level***
7:00 pm - 10:00 pm Dinner – **Chef Geoff's Downtown** (1301 Pennsylvania Ave)

Thursday, April 1, 2010

Day 2

*Day 2 Sessions are in the **Hemisphere B Room, Concourse Level** unless otherwise noted*
7:30 am - 8:30 am Continental Breakfast, Conference Registration and Discussion
DECISION SUPPORT - METRICS
8:30 am - 9:30 am Invited Stakeholder Panel – Joe Mitola, Stevens Institute of Technology, Panel Chair
- The Honorable Michael W. Wynne, USAF Ret.,
- Lieutenant General Robert J. "Bob" Elder Jr. USAF Ret.,
- Richard Hale, DISA
- Richard Marshall, Department of Homeland Security

9:30 am - 10:15 am *Discussion and* Break
10:15 am - 10:45 am The Utility of Security Standards - Jennifer Bayuk, Stevens Institute of Technology
10:45 am - 11:15 am Architectural Security Metrics - Georganne John, Analytic Services Inc.
11:15 am - 12:00 pm *Discussion*
12:00 pm - 1:00 pm *Lunch* – *Stevens Institute of Technology, Ground Level*
HOLISM - WORKFORCE
1:00 pm - 1:30 pm Security Process Modeling - Janet Carrier Oren, Stevens Institute of Technology
1:30 pm - 2:00 pm Pursuit of the Insider Threat - Paul Rohmeyer, Howe School of Technology Management, Stevens Institute of Technology
SOFTWARE COMPONENTS
2:00 pm - 3:00 pm Panel – Software Safety

A System Dependability Model and Accident Framework – Linda Laird, Stevens Institute of Technology
Software Engineering Security - Carol Woody, Carnegie Mellon University
Joint Software System Safety Engineering Handbook, Arch McKinley, Naval Ordnance Safety and Security Activity (NOSSA)
3:00 pm - 3:15 pm *Break*
ROADMAP REQUIREMENTS
3:15 pm - 4:30 pm Breakout sessions
Group 1 – Security Definition (Bayuk) - ***Hemisphere B***
Group 2 – Security Framework (Horowitz) - ***Stevens – Main Conference Room***
Group 3 – Security Metrics (Hamilton) - ***Stevens – Office A***
Group 4 – Security Workforce (Neuman) - ***Stevens – Office B***
4:30 pm - 5:30 pm Review of breakout sessions and workshop wrap-up – ***Hemisphere B Room, Concourse Level***
5:30 pm Workshop adjourns

Appendix D: References and Bibliography

References:

[1] MIL-HDBK-1785, "System Security Engineering Program Management Requirements," ed: US Department of Defense.

[2] President's Commission on Critical Infrastructure Protection, "Critical Foundations: Protecting America's Infrastructures," 1997.

[3] F. B. Schneider, Ed., *Trust in Cyberspace*. National Research Council, National Academy Press, 1999, p.^pp. Pages.

[4] INFOSEC Research Council (IRC), "Hard Problem List (1997-2005)," 2005.

[5] D. Maughan, "A Roadmap for Cybersecurity Research," US Department of Homeland SecurityNovember 2009 2009.

[6] Common Criteria Recognition Agreement, "Common Criteria for Information Technology Security Evaluation Version 3.1," ed, 2009.

[7] C. E. Irvine and K. Levitt, "Trusted Hardware: Can It Be Trustworthy?," presented at the Design Automation Conference, San Diego, California, USA, 2007.

[8] The Under Secretary of Defense for Acquisition Technology and Logistics and The Assistant Secretary of Defense for Networks and Information Integration DoD Chief Information Officer, "Report on Trusted Defense Systems," December 22, 2009 2009.

[9] R. Mogull, "An Open Letter to Robert Carr, CEO of Heartland Payment Systems," in *Securosis Blog*, ed, 2009.

[10] R. Dove, Ed., *The Interplay of Architecture, Security, and Systems Engineering* (INCOSE Insight 2). 2009, p.^pp. Pages.

[11] J. Boardman and B. Sauser, *Systems Thinking: Coping with 21st century problems*: Taylor & Francis, 2008.

[12] Y. Beres*, et al.*, "Using Security Metrics Coupled with Predictive Modeling and Simulation to Assess Security Processes," presented at the Third International Symposium on Empirical Software Engineering and Measurement, 2009.

[13] R. C.-W. P. Ahmad R. Amran, and David J. Parish, "Metrics for Network Forensics Conviction Evidence," in *International Conference for Internet Technology and Secured Transactions (ICITST)* 2009, pp. 1-8.

[14] H. Wang*, et al.*, "A Framework for Security Quantification of Networked Machines," presented at the 2nd International Conference on COMmunication Systems and NETworks, (COMSNETS), 2010

[15] L. Carin*, et al.*, "Quantitative Evaluation of Risk for Investment Efficient Strategies in Cybersecurity: The QuERIES Methodology," presented at the Metricon 3.0, San Jose, California 2008.

[16] B. F. Alshammari, Colin; Corney, Diane., "Security Metrics for Object-Oriented Class Designs," in *Ninth International Conference on Quality Software*, 2009 pp. 1550-6002.

[17] J. Bayuk*, et al.*, "White papers on Systems Security Definition, Framework, Metrics, and Workforce," presented at the SERC Security Workshop, Washington, D.C., 2010.

[18] International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC), "Systems and software engineering — Systems and software assurance — Part 2: Assurance case (ISO/IEC 15026)," ed, 2009.
[19] Networking and Information Technology Research and Development (NITRD), "National Cyber Leap Year Summit Report," 2009.
[20] M. Stanley Collins*, et al.*, "Engineering for System Assurance Version 1.0," National Defense Industrial Association System Assurance Committee, Ed., ed, 2008.
[21] F. Cohen*, et al.*, "A Preliminary Classification Scheme for Information System Threats, Attacks, and Defenses; A Cause and Effect Model; and Some Analysis Based on That Model," Sandia National LaboratoriesSeptember, 1998 1998.
[22] J. L. Bayuk and B. M. Horowitz, "An Architectural Systems Engineering Methodology for Addressing Cyber Security," *Submitted to Systems Engineering,* 2010.
[23] J. Bayuk, *Enterprise Security for the Executive: Setting the Tone at the Top*: Praeger, 2010.
[24] E. Chew*, et al.*, "Performance Measurement Guide for Information Security (Rev 1, first version 2003)," National Institute of Standards and Technology, Ed., ed, 2008.
[25] M. Swanson, "Security Self-Assessment Guide for Information Technology Systems," National Institute of Standards and Technology, Ed., ed, 2001.
[26] R. Ross*, et al.*, "Recommended Security Controls for Federal Information Systems, SP 800-53 Rev 2," National Institute of Standards and Technology, Ed., ed, 2007.
[27] H. Tipton and M. Krause, Eds., *Information Security Management Handbook, Sixth Edition, Volume 3*. Auerbach, 2009, p.^pp. Pages.
[28] J. Wirsbinski, "Systemic Security," PhD Dissertation, School of Systems and Enterprises, Stevens Institute of Technology, 2008.
[29] H. Shi and Y. Sakamoto, "Dynamic Coordination of Mobile Sensors through Competitive Learning," presented at the Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems, Toronto, Canada, 2010.
[30] L. Spizner, "The Honeynet Project," *IEEE Security & Privacy,* vol. 1, 2003.
[31] Y. Chen*, et al.*, "Detecting spoofing attacks in mobile wireless environments," presented at the Proceedings of the 6th Annual IEEE communications society conference on Sensor, Mesh and Ad Hoc Communications and Networks, Rome, Italy 2009.
[32] J. Cordasco and S. Wetzel, "An Attacker Model for MANET Routing Security," presented at the WiSec, Zurich, Switzerland, 2009.
[33] U. Meyer*, et al.* (2010, New Advances on Privacy-Preserving Policy Reconciliation. *Cryptology ePrint Archive*. Available: http://eprint.iacr.org/2010/064
[34] A. Jaquith, *Security Metrics*. Upper Saddle River, NJ: Pearson Education, 2007.
[35] Information Systems Audit and Control Association, "Control Objectives for Information Technology (COBIT)," ed. Rolling Meadows, IL: IT Governance Institute, 2007.
[36] L. A. Gordon and M. P. Loeb, *Managing Cybersecurity Resources*: McGraw-Hill, 2005.
[37] P. Herzog, "Open Source Security Testing Methodology Manual (OSSTMM) " ISECOM2010.
[38] Such as Intel® Trusted Execution Technology, see http://www.Intel.com/technology/security.
[39] G. Kim*, et al.*, *Visible Ops Security*: Information Technology Process Institute, 2008.
[40] S. Trimberger, "Trusted Design in FPGAs," presented at the Design Automation Conference, San Diego, California, USA, 2007.
[41] D. K. Holstein and K. Stouffer, "Trust but Verify Critical Infrastructure Cyber Security Solutions," presented at the 43rd Hawaii International Conference on System Sciences (HICSS), 2010

[42] T. Jiang and J. S. Baras, "Ant-based Adaptive Trust Evidence Distribution in MANET," presented at the Proceedings of the 24th International Conference on Distributed Computing Systems Workshops (ICDCSW'04) 2004
[43] R. Nilchiani, "Measuring Space Systems Flexibility: A Comprehensive Six-element Framework," *Systems Engineering,* vol. 10, p. 305, 2007.
[44] R. Anderson, *Security Engineering, Second Edition*: Wiley, 2008.
[45] J. Sherwood*, et al.*, *Enterprise Security Architecture*: CMP Books, 2005.
[46] Systems Engineering Research Center, "Body of Knowledge and Curriculum to Advance Systems Engineering (BKCASE)," ed. Stevens Institute of Technology and the Naval Postgraduate School, 2010+.
[47] M. J. Wheatley, *Leadership and the New Science, Discovering Order in a Chaotic World, Third Edition*. San Franciso, CA: Berrett-Koehler Publishers, Inc, 2006.
[48] R. Dove and L. Shirey, "On Discovery and Display of Agile Security Patterns," presented at the Conference on Systems Engineering Research, Stevens Institute of Technology, 2010.
[49] P. Langley*, et al.*, "Cognitive architectures: Research issues and challenges," *Cognitive Systems Research,* vol. 10, pp. 141–160, 2009.
[50] D. M. Buede, *The Engineering Design of Systems, Models and Methods*: Wiley, 2009.

Many influential studies have been cited throughout the text of this report and are listed as references above. In addition, these publications have helped to shape opinions on the topics described herein:

Checkland, Paul, *Systems Thinking, Systems Practice*, John Wiley & Sons, 1999.

Department of Defense, *Critical Program Information (CPI) Protection*. DoD Instruction (DoDI) 5200.39, 2008.

Defense Science Board, *High Performance Microchip Supply*, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, February 2005.

Defense Science Board, *Mission Impact of Foreign Influence on DoD Software*, Office of the Under Secretary of Defense for Acquisition, Technology, and Logistics, July 2007.

ISO/IEC DTR 15026-1, Systems and software engineering — Systems and software assurance, Draft January 2009.

ISO. International Organization for Standardization (ISO) and International Electrotechnical Commission (IEC). "Information technology — Security techniques — Code of practice for information security management (ISO/IEC 27002)," "Information technology — Security techniques — Information security management systems —Requirements (ISO/IEC 27002),", "Information technology — Security techniques — Information security risk management (ISO/IEC 27005)." from www.iso.org.

Joint Software Systems Safety Engineering Handbook (JSSSEHdbk), Ver. 2, 2010 Update (Original 1999).

NIST, National Institute of Standards and Technology (US), Special Publications on Security and Federal Information Processing Standards, guidance specifically referenced here is in SP-37, SP53, FIPS199, FIPS200 from csrc.nist.gov.

Suh, G.E. and S. Devadas, “Physical Unclonable Functions for Device Authentication and Secret Key Generation,” in Design Automation Conference, Association of Computing Machinery: San Diego, California, USA, June 4-8 2007.

Systems Security Engineering Handbook, MIL-HDBK-1885, US Department of Defense.

Tiri, K., “Side-Channel Attack Pitfalls,” in Design Automation Conference, Association of Computing Machinery: San Diego, California, USA, June 4-8 2007.

Trimberger, S., “Trusted Design in FPGAs,” in Design Automation Conference, Association of Computing Machinery: San Diego, California, USA, June 4-8 2007.